

Solutions for Rubik's Cube

识破天 2009.9 整理

目录

第零节 Cube Notion-----	III
第一节 层先法 Layer by Layer Method-----	1
第二节 棱先法 Edge-First Method-----	4
第三节 角先法 Corner first method-----	6
第四节 Ortega Corners-First Method-----	8
第五节 8355 Method-----	14
第六节 CFOP Method-----	17
第七节 CFCE-----	29
第八节 桥式解法 Gilles Roux Method-----	36
第九节 Petrus Method-----	45
第十节 Heise method-----	49
第十一节 ZZ method-----	53
第十二节 ZB Method-----	65
第十三节 VH Method-----	66
第十四节 Makisumi-Garron Last Slot (MGLS)-----	72
第十五节 Thistlethwaite method-----	73
第十六节 降群法简明教程-----	91
第十七节 My Corner-first Method-----	96
第十八节 Blindfold Cubing-----	100
第十九节 Commutators-----	114
第二十节 Mathematics of Cube-----	119
第二十一节 Cube Patterns-----	121
第二十二节 Bob Burton's Finger Tricks-----	131
第二十三节 Rubik's Revenge-----	138
第二十四节 Proffesor's Cube-----	142
第二十五节 V-Cube 7/6-----	144
第二十六节 Pocket Cube-----	145
第二十七节 粽子 Master Pyramorphix-----	150
第二十八节 五魔方 Megaminx-----	157
第二十九节 SQ1,SSQ1-----	160
第三十节 Skewb-----	185
第三十一节 Snake Cube-----	190
第三十二节 其它异形-----	192
附录 OLL/PLL complete Algorithms-----	196

声明

大部分资料来自互联网，部分由本人编辑改写，部分直接 copy，不再一一注明出处。版权所限，交流使用，勿做商业用途。

如有疑问，请联系 QQ: 229506994, Email: hegellogical@163.com。

第零节 Cube Notion

采用最为通用的 Harris 记法，Harris ENG notation:

六个面的基本转动								
	U	上层顺时针 90 度转动		R	右面顺时针 90 度转动		F	前面顺时针 90 度转动
	U'	上层逆时针 90 度转动		R'	右面逆时针 90 度转动		F'	前面逆时针 90 度转动
	U2	上层顺时针 180 度转动		R2	右面顺时针 180 度转动		F2	前面顺时针 180 度转动
	D	底层顺时针 90 度转动		L	左面顺时针 90 度转动		B	后面顺时针 90 度转动
	D'	底层逆时针 90 度转动		L'	左面逆时针 90 度转动		B'	后面逆时针 90 度转动
	D2	底层顺时针 180 度转动		L2	左面顺时针 180 度转动		B2	后面顺时针 180 度转动

中间层					
	E		M		S
	E'		M'		S'

	E2		M2		S2
--	----	--	----	--	----

整个魔方					
	x		y		z
	x'		y'		z'
	x2		y2		z2

魔方双层转

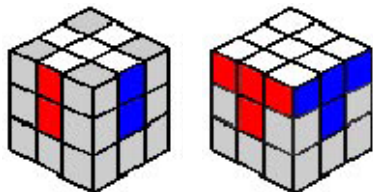
	u 上面两层作顺时针90度转(即转一下)		r 右面两层作顺时针90度转(即转一下)		f 前面两层作顺时针90度转(即转一下)
	u' 上面两层作逆时针90度转(即转一下)		r' 右面两层作逆时针90度转(即转一下)		f' 前面两层作逆时针90度转(即转一下)
	u2 上面两层作顺时针180度转而逆时针180度实际结果与顺时针是一样的!		r2 右面两层作顺时针180度转而逆时针180度实际结果与顺时针是一样的!		f2 前面两层作顺时针180度转而逆时针180度实际结果与顺时针是一样的!
	d 底面两层作顺时针90度转(即转一下)		l 左面两层作顺时针90度转(即转一下)		b 后面两层作顺时针90度转(即转一下)
	d' 底面两层作逆时针90度转(即转一下)		l' 左面两层作逆时针90度转(即转一下)		b' 后面两层作逆时针90度转(即转一下)
	d2 底面两层作顺时针180度转而逆时针180度实际结果与顺时针是一样的!		l2 左面两层作顺时针180度转而逆时针180度实际结果与顺时针是一样的!		b2 后面两层作顺时针180度转而逆时针180度实际结果与顺时针是一样的!

第一节 层先法 Layer by Layer Method

魔方入门级解法。有不少好教材，如 Jasmine Lee 的, Leyan 的, Joels 的等等。

一. 底层还原

可以先还原底层楞块（底面做十字），然后还原底层角块。



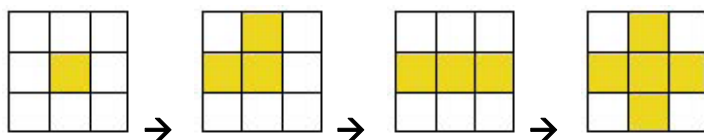
二. 中层楞块还原

第二层只剩下四个楞块，用到两个公式还原。注意，这一步对新手是一个坎，即放入新的块会对前面做好的面造成破坏。解决思路就是“配对+插入”。下面这两个公式实际上都是在做一件事：把楞块和它正下面的角块移到顶层配对，再一起插入下两层。理解了 this, CFOP 的 F2L 那一步 41 个公式就不需要死忆了。

U R U' R' U' F U F'	U' F' U F U R U' R'

三. 顶十字

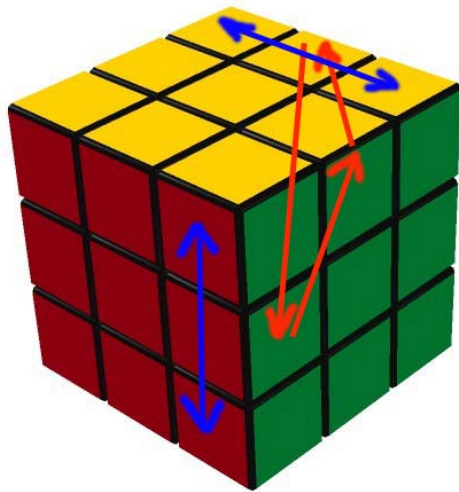
顶层楞块有四种情形，利用公式 **FRUR'U'F'**，使其转换成后续情形直至完成顶面十字。



FRUR'U'F'

注意，这是一个非常巧妙的公式。**RUR'U'**是个对易子，它等于楞块三轮换乘以两对角块对换：
(FR,UR,UB)(FRU,FRD)(UBR,UBL).见下图。

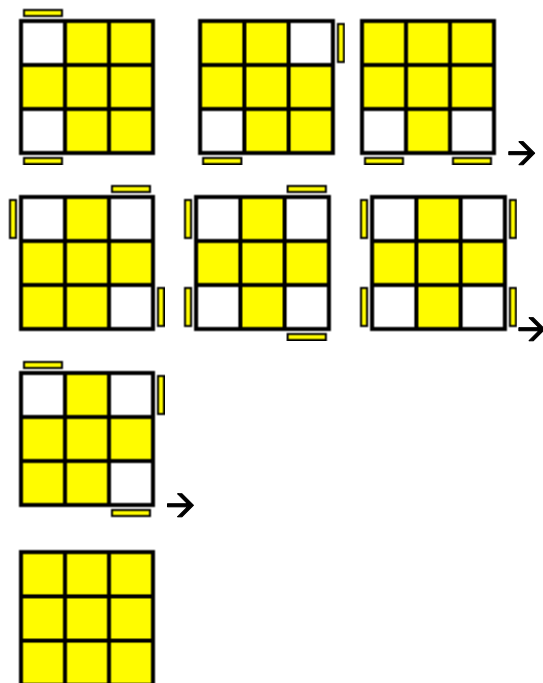
共轭变换 $F(\dots)F'$ 也设定的非常巧妙。考虑棱块方向，三轮换 (FR, UR, UB) 是对应楞块贴片的轮换， $[F]\{R\} \rightarrow [U]\{R\} \rightarrow [U]\{B\}$ 。[] 括号中的三个相应贴片轮换，{} 括号中的三个贴片轮换。这么轮换时考虑楞块方向，就明了了做十字时顶层为何这么摆放。



四. 顶面颜色还原

将四个顶角的 U 面翻到魔方的 U 面，其相对位置留待第五步。

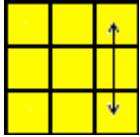
架好十字的顶层有七种状态，用一个公式 $RUR'URU2R'$ ，即可将顶层颜色全部还原。



$RUR'URU2R'$

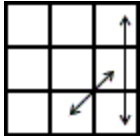
五. 顶角块位置还原

利用公式 $RU2R'U'RU2L'UR'U'L$ 对换两个角块 $URF-URB$ ，重复利用此公式还原所有角块位置。



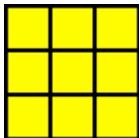
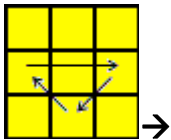
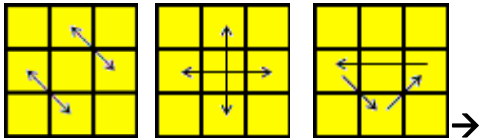
$RU2R'U'RU2L'UR'U'L$

这是 CFOP 方法的一个 PLL 公式，它同时对调了一对棱块的位置，对这一步没有影响，楞块位置留待下一步还原。



六. 顶楞块位置还原

还原顶层棱块有四种置换情形，利用三轮换公式 $F2UM'U2MUF2$ 转换至整体还原。



$F2UM'U2MUF2$

这个棱块三轮换公式是由三轮换 (UB,UF,FD) 共轭变换得到: $(F2U)-(M'U2MU2)-(U'F2)$.

第一排最后一个逆时针三循环可以利用 $M'U2MU2$ 的逆操作，一步到位: $(F2U)-(U2M'U2M)-(U'F2)$

第二节 棱先法 Edge-First Method

棱先法的思想是先还原 12 个棱块，再还原八个角块。

一. 还原底层棱

位置，颜色朝向都要还原

二. 还原中间层三个棱

预留一个棱位，用于和顶层棱进行棱置换。

三. 还原顶层四个棱和留下的中层棱

$F R' F' R$ 这个动作是使挨着右前上角的三个楞块逆时针轮换。 $F'R'FR$ 是使这三个楞块逆时针轮换。如下图示：



四. 还原角块

可以用合适的 Commutator 来角置换，角转向。

两个 3-cycle:

$R'D'R U' R'DR U$ ，轮换这三个角：RFU->LFU->RFD。如下图，同时这三个角的某三个面也轮换，（这三个角的其他面不轮换）。

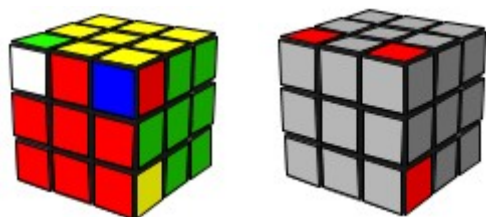
用三轮换把所有角块归位后，可能存在两个角块方向需要扭转，还是利用对易子实现。

a) F 面上逆时针三轮换 RFU->LFU->RFD

$P = R'D'R$ and $Q = U'$

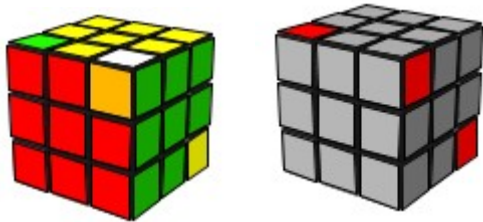
$R'D'R U' R'DR U$

对易子是 F Slice 上逆时针三轮换，RFU->LFU->RFD，这三个块的三轮换，实际上是三个块上的贴片之间的 3 个三轮换。



b) 三轮换

$P = ULU'$ and $Q = R2$, $ULU' R2 ULU' R2$ 使下面三个角轮换：

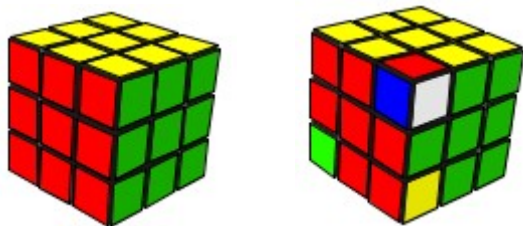


c) F 面上顺时针三轮换 $\mathbf{RFU} \rightarrow \mathbf{DFR} \rightarrow \mathbf{DLF}$

$$P = \mathbf{RUR}', Q = \mathbf{D}, P^{-1} = \mathbf{RU'R'}, Q^{-1} = \mathbf{D'}$$

$$\mathbf{PQP^{-1}Q^{-1}} = \mathbf{RUR' D RU'R' D'}$$

$\mathbf{RFU} \rightarrow \mathbf{DFR} \rightarrow \mathbf{DLF}$, 三个角块轮换时, 这三个角块上的三组个面也在做轮换。



注意, 例子中 P 本质上都是 $\mathbf{RUR'}$, $\mathbf{RU2R'}$ or $\mathbf{RU'R'}$ 这样的操作。任何角三轮换总可以在 12 步以内完成, 一般 10 步就可以。上面几个例子都是八步, 实现三轮换。如果不考虑角的朝向, 任何三个角都可以用上面前两种基本三轮换实现。假如考虑角块轮换后朝向也是正确的, 有的三个角不能通过上面基本轮换做到, 需要 **set-up**, 使它们可以应用基本三轮换。

Twist Two Corners

对易子可以用来同时扭转两个块的朝向。

$P = \mathbf{RU'R'URU'R'}$ and $Q = \mathbf{D}$. P 扭转逆时针 D 层上的 $\mathbf{RFD}$ 角, 不改变 D 层上任何其他方块。 $[P, Q]$ 扭转 $\mathbf{FDL}$, $\mathbf{FDR}$ 两个角, 一个顺时针一个逆时针。假如想顺时针扭转 $\mathbf{RFD}$, 逆时针扭转 $\mathbf{LDF}$ 用 P' 即可 $[P', Q]$.



如果取 $Q = \mathbf{D'}$ or $Q = \mathbf{D2}$, 那么可以我们扭转底面上其他两个角。

还可以使用 $P = \mathbf{F' U2 F R U2 R'}$ 扭转 $\mathbf{RFD}$ 角, 再取 $Q = \mathbf{D}$ 。注意这是 8355 方法里面扭转角块用到的方法。

第三节 角先法 Corner first method

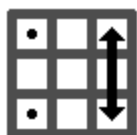
角先法有 Waterman Method, Jelinek Method, Ortega Method, 大概步骤一样, 有一些小的差异。Ortega Method 在还原底层顶层颜色时都不考虑位置, 调整角块位置时八个角块一块调整有六个算法, 还原楞块方向有 12 个算法, 速度要快得多。

一. 底面四角归位

颜色, 相对位置都要正确

二. 调对顶层角块相对位置

有下面两种情况。



$L U' R' U L' U' R U^2$



$L U^2 L' R' U L U' R U^2 L' U$

三. 顶层颜色朝上

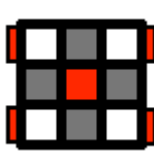
先将顶层角块的顶层颜色全部调整朝上, 第三步在调整顶层角块相对位置。有七种情况, 七个公式。



$R U R' U R U^2 R' U^2$



$L' U' L U' L' U^2 L U^2$



$R U R' U R U' R' U R U^2 R'$



$R U^2 R^2 U' R^2 U' R^2 U^2 R$



Execution of 1. algorithm transforms to the situation in the picture



Execution of 1. algorithm transforms to the situation in the picture



Execution of 1. algorithm transforms to the situation in the picture

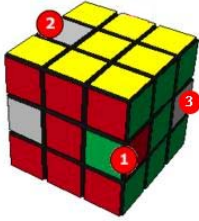
四. 还原顶层三个棱块

五. 翻转魔方底层朝上, 还原四个棱

利用第四步预留的顶层（现为底层）那个棱做转换, 而不改变其他三个顶棱块位置。

六. 调整中层棱朝向

反转顶底，缺一棱的朝上。判断中层棱朝向方法很简单：就是看该棱某一面的颜色与其所在面或相对面中心块的颜色是否一致，一致的话就正确，不一致就错误。调整棱方向的方法如下：



公式 6-1: **R MF R2 MF' R** 他的变体 6-2: **R' MF R2 MF' R'**

这两个公式调整棱块朝向。

运用公式 6-1，使得 1 号块转到 2 号位置，2 号块转到 3 号位置，3 号块转到 1 号位置，同时保持所有其他块不动。即 132 块轮换为 321。

运用公式 6-2，使得 1 号块转到 3 号位置，3 号块转到 2 号位置，2 号块转到 1 号位置，同时保持所有其他块不动。即 132 块轮换为 213。

这个轮换过程是为了改变了棱块 2 颜色朝向。转动中间层，使得 L 或 R 面中心块颜色与 2 号块侧面(L 面)颜色相同，2 号块轮换到 1 或 3 位时其颜色朝向即正确。

中层棱朝向都正确后，判断并采用这两个公式之一即可把顶层棱块归位。

七. 中层棱归位

1. 中层三个楞块位置不正确，**F2 MU F2** 使之归位。这一步之前要观察三个楞块相对位置，或转动中间层，或翻转魔方，使三个楞块处于合适位置。
2. 中层四个楞块位置不正确，用一次公式 **F2 MU F2**，即削减成情况 1。

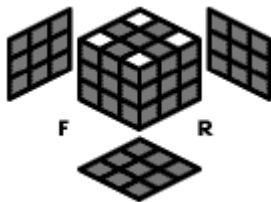
第四节 Ortega Corners-First Method

By Victor Ortega and Josef Jelinek

This solution method orients cubies before positioning them. The idea is that it is easier to permute cubies after they've been oriented than before orienting them, because once the cubies have been oriented, the facelet colors that determine their permutation make easily identifiable patterns on the cube. Orienting cubies, whether done before or after positioning them, is always easy because orientation requires focusing on only one face color and on the patterns that that color makes on the cube. For middle-slice edges on the last layer, permuting cubies after they've been oriented is a very simple affair, thus reinforcing this principle.

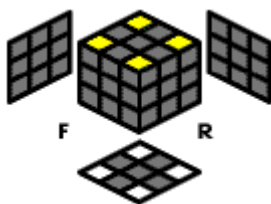
Stage I: Solve the corners

1. Orient U corners



You should be able to manage this on your own. Don't worry about positions--all corners will be permuted in step 3. You should be able to orient U corners in 6 moves or less. For the greatest speed and efficiency, try to do this in one look.

2. Orient D corners



Rotate the whole cube so that D becomes U. Orient the corners depending on which of the seven patterns below you see:



letter T pattern:

$R \ U \ R' \ U' \ F' \ U' \ F$



letter L pattern:

$F \ R' \ F' \ U' \ R' \ U \ R$



sune pattern #1:
 $R\ U^2\ R'\ U'\ R\ U'\ R'$



sune pattern #2:
 $R\ U\ R'\ U\ R\ U^2\ R'$
 (inverse of #1 in both respects)



letter pi pattern:
 $R\ U\ R^2\ F'\ R^2\ U\ R'$



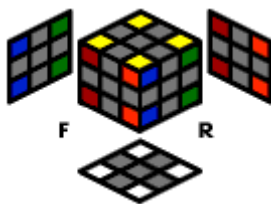
letter U pattern:
 $R'\ F'\ U'\ F\ U\ R$



letter H pattern:
 $R^2\ U^2\ R'\ U^2\ R^2$

3. Permute all corners, by method of number of solved "edges"

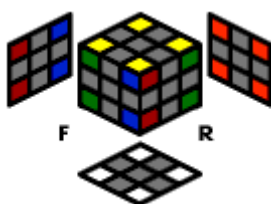
If two adjacent corners are positioned correctly relative to each other, it's a right pair. Count the right pair and proceed with one of the following sequences:



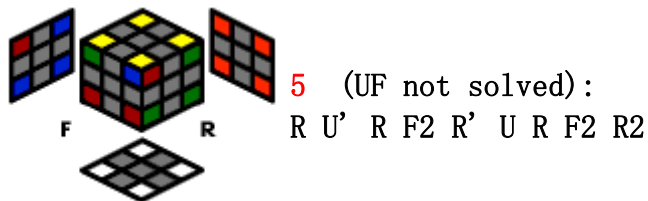
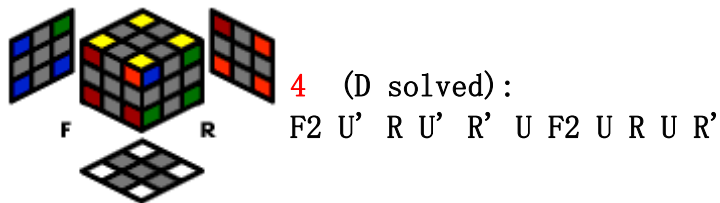
0:
 $F^2\ R^2\ F^2$



1 (DB solved):
 $R\ U'\ F\ U^2\ F'\ U\ R'$
 ((UB solved): $R'\ U\ R'\ B^2\ R\ U'\ R$)



2 (DB and UB solved):
 $R^2\ U\ F^2\ U^2\ R^2\ U\ R^2$

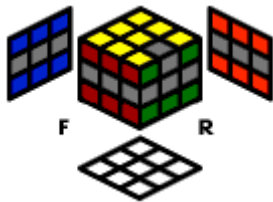


Stage II: Solve the edges

4. Solve three U edges

5. Solve three D edges

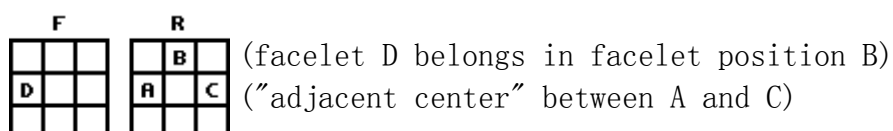
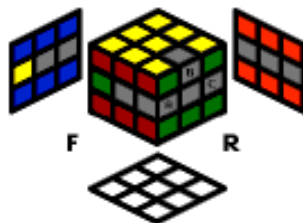
6. Solve one more U or D edge, depending on which is easier

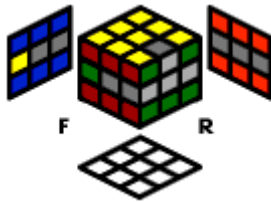
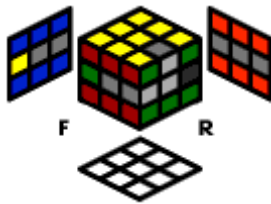
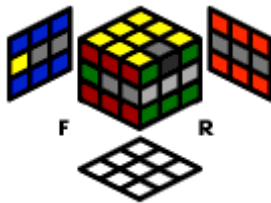
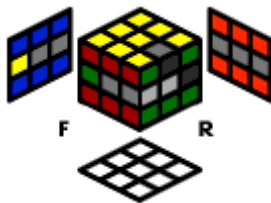
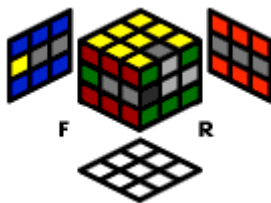
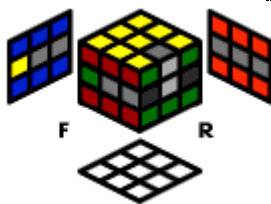


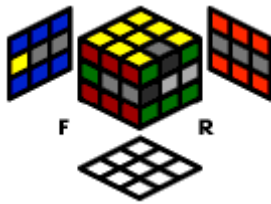
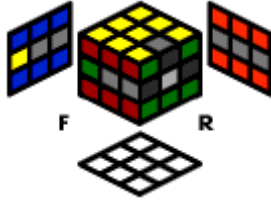
7. Solve last U edge, orient middle layer

a) U edge in the middle layer

In the diagram below, edges A, B, or C are oriented correctly (C) if that facelet's color matches the adjacent center or the opposite center. Otherwise it's incorrectly oriented (I).

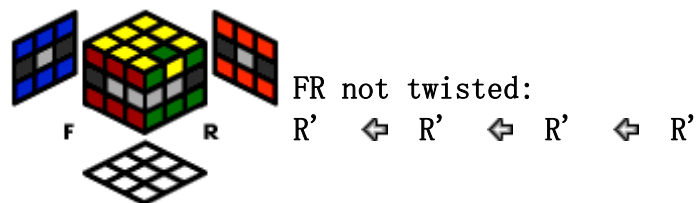
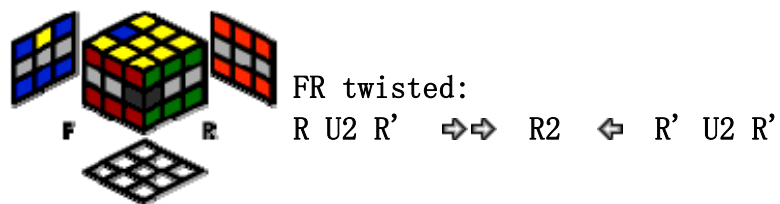


Edge A	Edge B	Edge C	Pattern	Sequence
C	C	C		$\Rightarrow R \Rightarrow R' \Leftarrow R' \Rightarrow R$
C	C	I		$\Rightarrow R' \Leftarrow R' \Rightarrow R' \Leftarrow R'$
C	I	C		$\Leftarrow R \Rightarrow R^2 \Rightarrow R$
C	I	I		$R' \Leftarrow R \Rightarrow R' \Rightarrow R$
I	C	C		$R^2 \Rightarrow R \Leftarrow R \Rightarrow R \Leftarrow R'$
I	C	I		$R' \Rightarrow R' \Leftarrow R' \Leftarrow R \Rightarrow R$ $\Leftarrow R$

I	I	C	 $R' \leftarrow R' \rightarrow R \rightarrow R$
I	I	I	 $R' \rightarrow R' \rightarrow R' \leftarrow R^2 \leftarrow R$

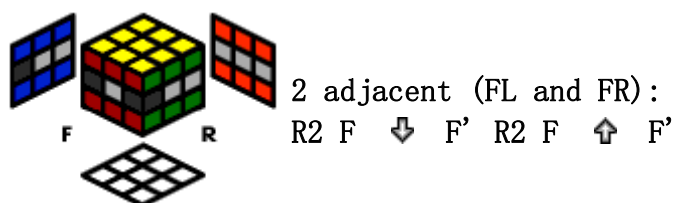
b) U edge in position but twisted

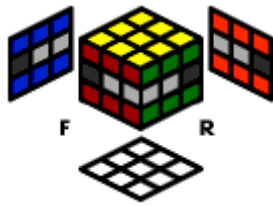
There will be 1 or 3 twisted edges in the middle layer:



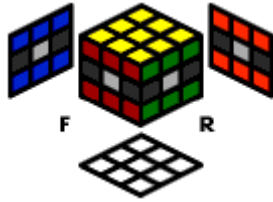
c) U edge solved

There will be 0, 2, or 4 edges twisted in the middle layer:



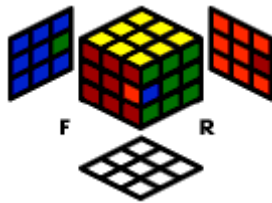


2 opposite (FL and BR):
 $F \downarrow F' R^2 F \uparrow F' R^2$

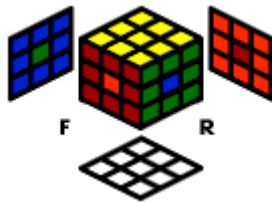


all four:
 $R F^2 R^2 \Rightarrow R \Rightarrow R F^2 R^2 \Rightarrow R$
 all four, when all edges are positioned correctly:
 $F' U' F (\Rightarrow R)^4 F' U F$

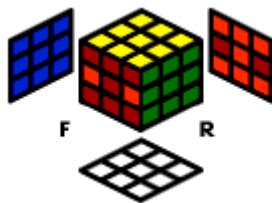
8. Position middle layer



send FR to BL, BL to BR, and BR to FR:
 $R^2 \Leftrightarrow R^2 (\Rightarrow)$



exchange centers with opposites:
 $\downarrow\downarrow \Leftrightarrow \downarrow\downarrow (\Rightarrow)$



exchange FR with BR, FL with BL:
 $R^2 \Rightarrow R^2 (\Rightarrow\Rightarrow)$

第五节 8355 Method

Wikipedia “How to Solve the Rubik's Cube” 介绍的玩法，略作改变。

8355 法，又称 Working Corner/Edge Method, 或最小记忆玩法。由台湾许技江老师发现。一二两步，许技江的原始做法是先还原底层 8 个块，再还原第二层 3 个楞块。第三步还原剩余 5 个楞块，最后还原剩余 5 个角块。故称 8355 玩法。

一. 底面十字

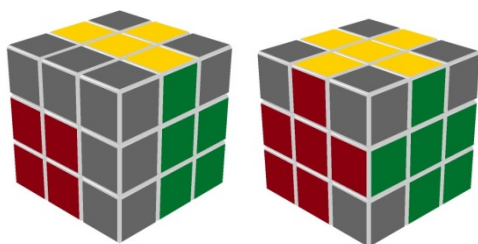


二. 同时还原底面角块以及对应的第二层楞块，空留下一对（右后位） CFOP 的 F2L 方法



三. 还原所有剩余棱块（5 个）

把魔方整体转一下，空缺的 F2L 放在右前方，便于观察。
这一步我们分成两个阶段，



第一步很简单。
第二步三个公式：

	两个棱色块位置正确，朝向需要翻转 $R\ U'\ R'\ U\ F'\ U\ F U'$
	两个棱块 R 面都处于魔方的 R 面，但位置需要互换 $U'\ F'\ U'\ F\ U'\ F'\ U'\ F U'$
	两个棱块方向要翻转，位置需要互换 $R U\ R'\ U\ R\ U\ R'\ U^2$

$R-U$ (or U') $-R'$, **$F'\ U'\ F$** 这两个基本动作之后，任意一个可调整三个相邻楞块的相对位置。熟悉之后第三步这几个公式都不需要，顺手轮换这五个块即可。

四. 摆好剩余角块的位置，不考虑颜色朝向

这一步为了便于操作，再把魔方整体转一下，把空缺的 F2L 放到右后方。

这一步三个基本动作是 $\{U, R, B\}$ 。旋转顶面，用 $L\ D^2\ L'$ 这一操作调换右下后和前上左两个角块。（当然，也可以用它的镜像算法 $F'\ D^2\ F$ 达到同样目的。）

首先观察右下后处角块的颜色，它应该转到对应顶角位置。然后旋转顶面让对应顶角位置到达前上左角。举个具体例子来说明，魔方上黄下白左绿右蓝前橙后红，右下后角块黄红绿块，则旋转顶面，使得顶面黄绿，黄红两个棱块夹得那个角块位置旋到左前上角， $L D^2 L'$ 即把右下后角的蓝红绿块归到正确的顶角位。

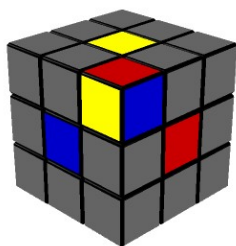
重复上述步骤，直到四个顶角一个底角都被还原到正确位置。调换角块时，魔方其它部分被打乱了，当角块归位时打乱的那些块自动复原。

假如右下后角块不是顶层角块，一次 $L\ D^2\ L'$ 即把左前上角位置的 U 面角块调到右下后位，再用上述方法使五个角块还原。

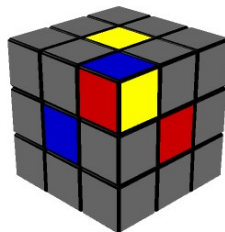
五. 调整五个角块颜色朝向

这步很简单，还是用 $R'D^2R$ 和 FD^2F' 。首先你找到一对角色块，他们要处于同一个面上，而且一个要顺时针翻转，一个要逆时针翻转。

所谓顺时针翻转，逆时针翻转，如下图所示：



角块需要顺时针翻转



角块需要逆时针翻转

Case 1. 两个角块一个顺时针一个逆时针

把这两个角色块放在同一个面顶面上，这时黄色面可以不一定朝上。找到那个逆时针的角块，旋转顶面让它到达右前上角，应用 $R'D2R FD2F'$ ，这时这个角块方向就扭转好了（魔方被有规则的打乱了）。然后，找到那个需要顺时针扭转的角块，旋转顶面让它到达右前上角，应用刚才算法的逆算法 $FD2F' R'D2R$ ，它的方向也扭转正确了。魔方打乱部分也自动复原了。

如果你先翻转那个顺时针的再扭转那个逆时针的，就先用 $FD2F' R'D2R$ 再用 $R'D2R FD2F'$ 。总之这两个算法要配对使用。

Case 2. 三个角块都是一个时针方向

也有的情况你要翻 3 个角，他们都要顺时针或逆时针翻转的，比如三个角 ABC 都是逆时针的，就先取两个 AB 放在顶面上，A 转到上右前角用一次 $R'D2R FD2F'$ ，其方向就正确了。再把 B 转到右前角，做一次 $FD2F' R'D2R$ ，就会变成顺时针，（逆时针的角顺时针转一次就变成顺时针），这时 BC 一顺一逆，再按照 Case 1 来还原。

加速方案

此文是为了不记公式尽量简化的方法，如果为了提速，可以合并第四第五步，一步复原角块。见许技江网站。

第六节 CFOP Method

流行的速度玩法，119 个公式。

一. 底面十字

二. F2L (first 2 layers): 直接把前两层还原

不需要记公式，掌握其技巧即可。以还原前右下角块和楞块为例，目标是此位置楞块角块调整到一起，同时归位。过程中有两个技巧：

1. 将角块转动右前上位，先 F 或 R，再 U 或 U' 改变其颜色朝向。
2. 将角块转动到左前上位或右后上位，再 F' (或 R)，将其调离顶层，再转动顶层调整楞块至合适位置，再 F(或 R') 使角块回到顶层与楞块紧靠作为一个整体归位。

三. OLL(orientation of last layer)调整顶面颜色向上

需要记公式，见下表。

四. PLL(permutation of last layer)调整顶面顺序

调整角块边块位置，使归位。

公式见下表。

Fridrich Method Tutorial

Shotaro Makisumi (Macky)

Invented by [Jessica Fridrich](#) in the early 1980s, the Fridrich Method gained popularity among the second generation of speedcubers when it was published online in 1997. By far today's most popular speedcubing method, it is used by most world-class speedcubers with minor modifications.

Step	Description	Moves (Average)	Time (Average)
Cross	The four edges of one layer are placed. (In the applet to the right, the cross has been made on the bottom layer.)	8	2 sec
First Two Layers (F2L) (Advanced F2L)	Completes the first two layers by fixing the four corner-edge pairs between the cross edges in four steps, one slot at a time. 41 standard patterns counting mirrors for corner-edge pairs.	$7 \times 4 = 28$	7 sec
Orientation of Last Layer (OLL) printable page applets page	Corrects the orientation (flip) of all last layer pieces in one step so that every piece has the last layer color on top. 57 patterns counting mirrors and inverses.	9	2 sec
Permutation of Last Layer (PLL) printable page applets page	Corrects the permutation (placement) of all last layer pieces in one step. 21 patterns counting mirrors and inverses.	11	2.5 sec
Total	41 F2L and 78 last layer algorithms to memorize	56	13.5 sec

Cross

The goal of this step is to form a cross on the bottom layer by correctly placing the four edge pieces around the bottom center. Unlike the other steps of the Fridrich method, the cross is solved almost entirely intuitively. You should try to plan your cross solution in the 15 seconds of inspection given before each solve.

A key fact to understand about the cube is that it consists not of 54 independent stickers, but rather of 8 corner pieces (3 stickers each) and 12 edges (2 stickers each), together called the cubies. Because stickers always remain on the same cubie, it is not enough to place the blue *stickers* on the blue face. We must make sure that each edge *cubie* is in its correct spot.

First Two Layers (F2L)

After cross comes the F2L (First Two Layers), arguably the most important step of Fridrich Method. This step completes the first two layers by fixing the four corner-edge pairs (**slots**) between the cross edges in four steps, one slot at a time. There are 41 basic "algorithms" to insert a corner and its corresponding edge into their correct spot in the correct orientation. However, most of these are short and intuitive compared to the last layer algorithms.

Because the four pairs can be solved in any order, F2L is a relatively flexive step and allows for many time-saving techniques. First take a look at [Getting Fast with an Easy Method](#) for such techniques in the simpler setting of layer-by-layer. Also, rather than blindly applying the algorithms, it is important to understand how each one works. Consider learning the F2L intuitively first from [Doug Reed's guide](#) and then comparing your algorithms with my list or with those at [cubeloop.com](#).

This page first lists the 41 standard F2L cases with target slot FR. Each is written as I would perform it, making ample use of double layer and whole cube turns. Where appropriate, I have provided multiple algorithms. While one per case will suffice, one algorithm is often better than the other from a certain angle. The rest of the page explains some techniques that require no extra algorithm. To really master F2L, I recommend that you work on these techniques at the same time as you learn the algorithms.

More advanced techniques, requiring new algorithms, are covered in [Advanced F2L](#).

Orientation of Last Layer (OLL)

The first of the two-step last layer and the least important of all steps in the Fridrich method, OLL corrects the orientation of all last layer pieces in one step so that every sticker on top has the top layer color. Permutation is usually not preserved.

These are my current OLL algorithms with pattern codes from JSCC. The picture represents the top face of the cube after F2L, where a gray square or a black bar on the side indicates a sticker with the last layer color. The algorithm is applied after placing the last face on top. See the [Applet Page](#) if this doesn't make sense. Most of these algorithms have been taken or are based on other cubers' algorithms. Try the OLL algorithms in [these sites](#) to see which one you like best.

Permutation of Last Layer (PLL)

PLL, the last step of the [Fridrich Method](#), permutes the last layer pieces without disturbing their orientation. There are 21 patterns including mirrors and inverses, much less than for OLL. However, on average a PLL is longer than an OLL. Go as fast as you can on PLL. Your initial goal should be sub-2.5 sec for 1 execution (sub-10 for 4 consecutive executions) for each pattern.

Here you can find the algorithms I use. I will probably continue to update this list as people discover better algorithms. Do check out [other cubers' lists](#) and find your best match. Since PLL is often where you can personalize the fingering, try different ways to perform the same

algorithm by chunking into different substeps or by changing when to regrip. Keep in that you don't need to worry about whole cube turns and that you need to put down the cube to stop the timer.

Cross



R'



R^2



$R F R^2$



$R F R$



$R' F R$



$R' F R' B' R^2$



$R' F R'$



$R' F R' B'$



$R F R'$



$R F B' R'$

F2L

A1



U R U' R' U' F' U F

A2



U' F' U' F U R U' R'

B1



F2 U' F2 U' F2 U2 F2

B2



R U' R' F' U2 F

C1



R U2 R U' R' U R U2 R2

C2



R2 U2 R' U' R U' R' U2 R'

D1



F' U' F' U' L' U L F2

D2



R U' R U B U' B' R2

E1



F' U F2 R' F' R

E2



R U' R2 F R F'

F1



U F' U F R U R'

F2



U' R U' R' F' U' F

G1



U F' U F U2 F' U F

G2



U' R U' R' U2 R U' R'

H1



U F' U' F U' R U R'

H2



U' R U R' U F' U' F

I1



U' F' U F

I2



U R U' R'

J1



U F' U2 F U' R U R'

J2



U' R U2 R' U F' U' F

K1


 $R U' R' U^2 F' U' F$

K2


 $F' U F U^2 R U R'$

L1


 $U' R U' R' U R U R'$

L2


 $U F' U F U' F' U' F$

M1


 $R U R B' R' B U^2 R'$

M2


 $F' U' F' L F L' U^2 F$

N1


 $F' U^2 F U F' U' F$

N2


 $R U^2 R' U' R U R'$

Q1


 $U F' U' F U^2 F' U F$

Q2


 $U' R U R' U^2 R U' R'$

R1


 $U F' U F U' R U R'$

R2


 $U' R U' R' U F' U' F$

S1


 $U F' U^2 F U^2 F' U F$

S2


 $U' R U^2 R' U^2 R U' R'$

T1


 $R U R'$

T2


 $F' U' F$

U1


 $U' F' U^2 F U F' U F$

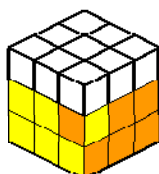
U2


 $U R U^2 R' U R U' R'$

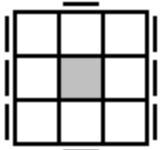
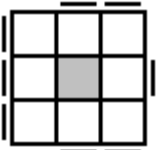
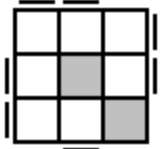
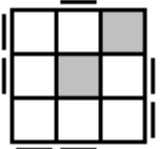
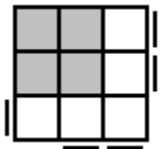
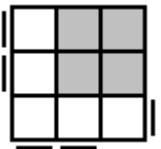
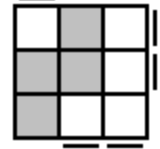
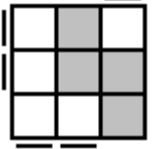
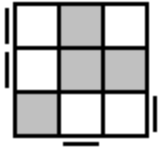
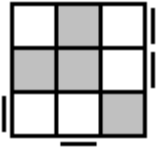
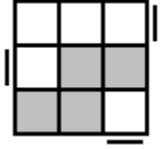
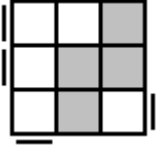
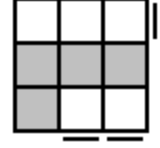
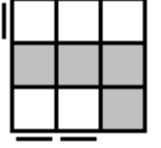
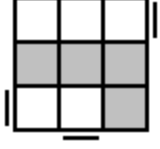
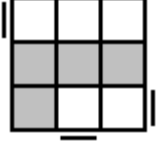
V1

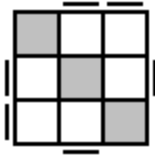
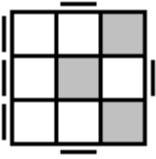
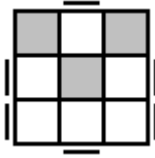
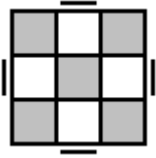
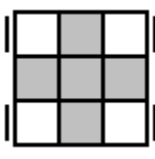
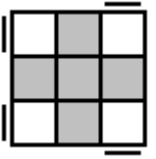
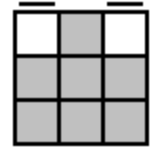
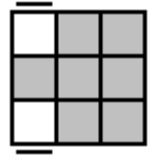
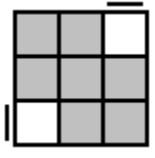
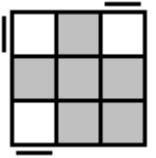
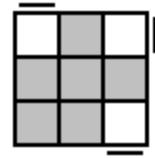
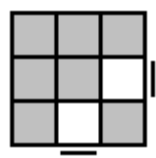
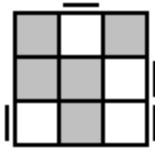
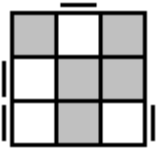
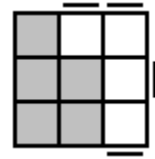
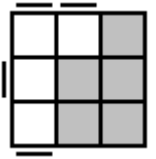

 $U^2 R U R' U R U' R'$

V2

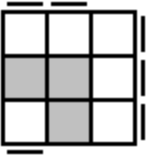
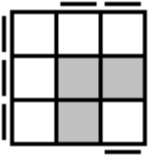
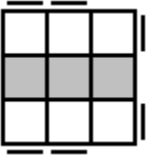
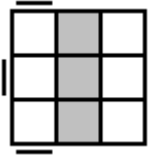
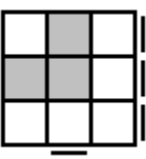
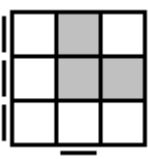
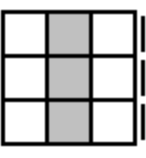
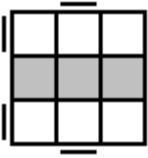
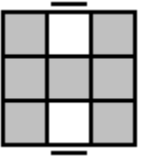

 $U^2 F' U' F U' F' U F$

 $R U' 2 R' U R U' 2 R' U F' U' F$

OLL

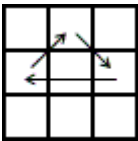
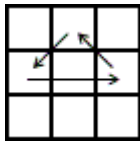
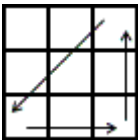
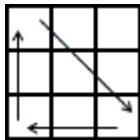
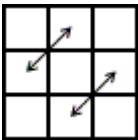
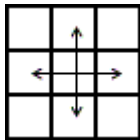
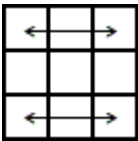
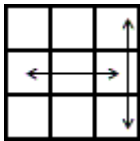
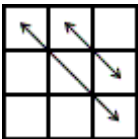
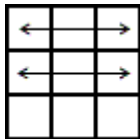
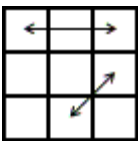
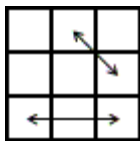
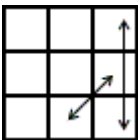
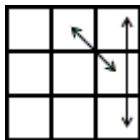
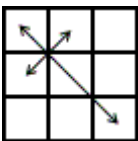
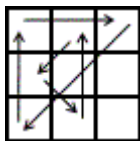
1 	$RU2-R2'FRF'U2'-R'FRF'$	2 	$F-RUR'U'-F'-Fw-RUR'U'-F'w$
3 	$R'RwU-Rw'U2-RwU-R'UR2Rw'$	4 	$R'U2x-R'URU'y-R'U'R'UR'F$
5 	$Lw'U2'LUL'ULw$	6 	$RwU2'R'U'RU'Rw'$
7 	$RwU-R'URU2'Rw'$	8 	$Lw'U'-LU'L'U2Lw$
9 	$F'U'F-LF'L'-U-LFL'$	10 	$FUF'-R'FR-U'-R'F'R$
11 	$L'R2BR'BR-B2R'BR'L$	12 	$F-RUR'U'-F'-y-F-RUR'U'-F'$
13 	$LF'L'-U'-LFL'-F'UF$	14 	$R'FR-U-R'FR-FU'F'$
15 	$L'B'L-R'U'RU-L'BL$	16 	$RBR'-LUL'U'-RB'R'$

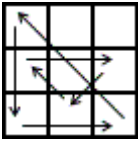
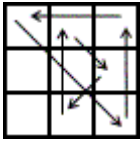
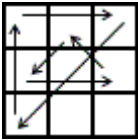
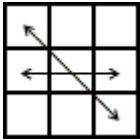
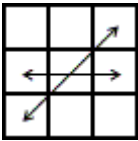
17 	RUR'U-R'FRF'U2-R'FRF'	18 	RU2-R2'FRF'-U2'RwR'URU'Rw'
19 	MU-RUR'U'Rw-R2'FRF'	20 	MU-RUR'U'-Rw2R2'URU'Rw'
21 	RU-R'URU'R'UR-U2R'	22 	RU2'-R2'U'R2U'R2'-U2R
23 	L2wU'R-D2-R'UR-D2R	24 	RwUR'U'Rw'FRF
25 	Lw'U'L'U-RU'LUX'	26 	L'U'LU'L'U2L
27 	RUR'URU2R'	28 	RwUR'U'R-Rw'URU'R'
29 	B'R'B-L'-B'R2B'-R'B2L	30 	BLB'-R-BL2B-LB2R'
31 	L'D'w-RDw-LU'L'B'L	32 	RDw-L'D'w-R'URBR'

33	RUR'U'-R'FRF'	34	RUR'U'(x)-D'-R'UR-E'
35	RU2-R2'FRF'-RU2R'	36	R'U'RU'R'URU-RB'R'B
37	B'RBR'-U'R'UR	38	LUL'ULU'L'U'-L'BLB'
39	LF'L'U'LUFU'L'	40	R'FRUR'U'F'UR
41	RU'R'U2RU(y)RU'R'U'F'	42	L'ULU2L'U'(y)L'ULUF
43	F'wL'U'LUFw	44	FwRUR'U'F'w
45	FRUR'U'F'	46	RU-RB'R'B-U'R'
47	F'-L'U'LUL'U'LU-F	48	F-RUR'U'RUR'U'-F'

49 	LwU'Lw2'ULw2ULw'2U'Lw	50 	Rw'URw2U'Rw2'U'R2wURw'
51 	F-URU'R'URU'R'-F'	52 	R'U'RU'R'-Dw-R'URB
53 	Lw'U'- LU'L'ULU'L'U2-Lw	54 	RwU-R'URU'R'URU2-Rw'
55 	RU2'-R2'U'RU'R'U2'-FRF'	56 	F-RUR'U'RF'-RwUR'U'Rw'
57 	RUR'U'-Rw-R'URU'Rw'		

PLL

1 	R'UR'U'-R'U'-R'URUR2	2 	R2U'R'URURURU'R
3 	RB'R-F2-R'BR-F2R2	4 	L'BL'-F2LB'L'-F2L2
5 	UR'U'RU'RU R U'R'U RU-R2 U'R'U	6 	M2UM2U2M2UM2
7 	xUR'U'LURU'-R2w'U'RULU'R'U	8 	RUR'U'R'FR2U'R'U'RUR'F'
9 	R'UR'U'-y-R'F'R2U'R'UR'FRF	10 	R'URU'R2'F'U'F-URU'x'R2U'R'U
11 	R'U2-RU2'-R'FRUR'U'-R'F'R2U'	12 	RU2'R'U2-LwU'L'wU'-RULwUL'wR'U
13 	RU2R'U'RU2L'UR'U'L	14 	R'U2RUR'U2LU'RUL'
15 	FRU'R'URUR'F'-RUR'U'R'FRF'	16 	RUR'y'R2U'wRU'R'UR'UwR2

17 	$R2'Uw'RUR'-UwR2FwR'Fw'$	18 	$L'U'LyL2UwL'ULU'LU'wL2$
19 	$R2Uw-R'UR'U'RU'w-R2'B'wRBw$	20 	$R'UR'FRF'RU'-R'F'UF-RUR'U'R$
21 	$(LU'RU2L'UR')*2U'$		

第七节 CFCE

CFCE 方法也称 Guus Razoux Schultz's method (Guus 是 82 年匈牙利世界魔方竞标赛亚军). CFCE 与 CFOP 方法接近, 区别在于后两步。

OLL 只求顶面同色, 暂时不管各块的位置; 接着 PLL 只求保持顶面同色的条件下调整各块的位置。

CLL 不管棱块, 把角块一部还原, 即位置和颜色朝向同时还原; 接下来对于棱块, 用 ELL 方法, 把位置和颜色朝向同时还原。

两者都是把顶层任务分解为两个小任务, 但是具体方式很不同, 各有特点。这一节来介绍 CFCE 方法。

一. Cross in first layer

二. First 2 Layers

三. Corners of Last Layer

一次性还原四个角块的方向和位置 共 42 个公式

调整 U 面, 在下表中找到与之匹配的情形	 A	 B	 C	 D
 1	I	R'U'R'U2R	RUR'URU2R'	yRUR'URU'R'URU'R'URU2R'
 2	R'UL'U2RU'R'U2RL	yRU2R'-L'U2RU'R'U2R-LU2R'	L'U2LU2LF'L'F	FU2F'U'-R'FRU'R'F'R
		(C4)'	(B3)'	(E5)'
 3	[U'] A2	B'RB'R'U2R'U2R	FR'F'RU2RU2R'	L'URU'L'-UR2U2RUR'UR
		(C2)'	(B4)'	(E3)'

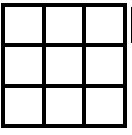
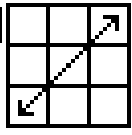
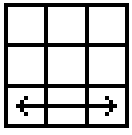
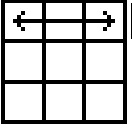
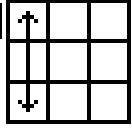
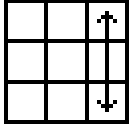
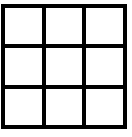
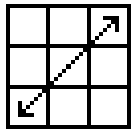
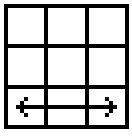
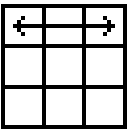
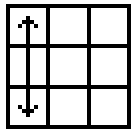
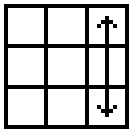
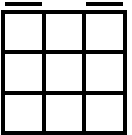
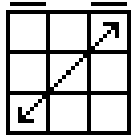
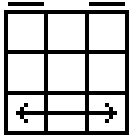
 4	[U2] A2	LU2L'U2L'BLB' (C3)'	yRU2L'-R'U2RUR'U2L-RU2R' (B2)'	y'FR'F'RURU'R' (F5)'
 5	[U] A2	R'ULU'RUL' (C5)'	RU'LUR'U'R (B5)'	F'LFL'U'L'UL (F3)'
 6	(R'UL'U2RU'L)2	RU2R'FR'F'RU'RU'R' (C6)'	R'U2RB'RBR'UR'UR (B6)'	RUL'R'U2RUR'U2L

续上表	 E	 F	 G	 H
 1	R'U'RUR'U2-R2-UR'URU2 R' (F1)'	FR'FR2U'R'U'RUR'F 2 (E1)'	FRUR'U'RUR'U'F' R'U2R2UR2UR2U2R' (G4)'	RUR'URU'RUR'URU2 (H4) (G3)'
 2	yRU'L'UR'U2yR'URB	R'URU2R'L'URU'L (F2)'	R'U2RUR'U'RU2RB'R'B (G4)'	[U2] H4 (H4) (G3)'
 3	R'U2RU'LU2R'URU2L' (D3)'	F'U'FUFR'F'R (D5)'	[U']F-RU'R'U'RUR'URU'R'- F' (H2)' (H4)'	RUR'URUL'UR'U'L (H5) (G5)'
 4	F-RU'R'URUR'URU'R'-F' (F4)'	FR'U'R2U'R2U2RU' F (E4)'	RU2R'U'RUR'U2R'FRF' (G2)'	F-RUR'U'RUR'URUR'-F' (H2) (G3)'

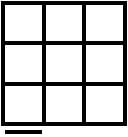
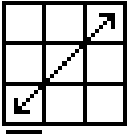
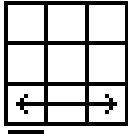
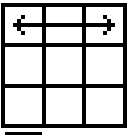
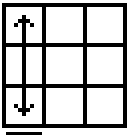
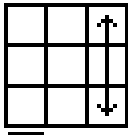
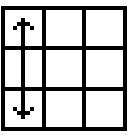
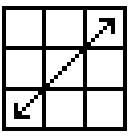
 5	R2D'RU2R'DRU2R (D2)'	y'RUR'U'R'FR' (D4)'	RU'L'UR'ULUL'UL (H3)' (H5)'	[U2] H3 (H3) (G5)'
 6	LFUF'U'L'	F'UFURU2R'FR'F'R (F6)'	y'FUF'UFU'RU'R'F' (G6)'	y'F-RUR'U'RUR'U'RUR'U'- F'

下面是来自 speedcubing 的算法表格，另一种表示方法。

(<http://www.speedcubing.com/finallayer.html>)

 code: c1 FR'FR2U'R'U'RUR'F2 (11,13) RU2R2F2D'F'DF'R2U2R' (11,16)	F F .. B B	 code: c2 F2R2FU'L'ULF'R2F2 (10,14) B2L2BR'U'RUB'L2B2 (10,14)	F B .. F B	 code: c3 F'L2F'R2FL2F'R2F2 (9,14)	F F .. L R
 code: c4 R'URU2R'L'URU'L (10,11) FRF'LFR2FRF2L' (10,12)	R L .. F F	 code: c5 FUF'U'F'LFL' (8,8) R'F'RB'R'FRB (8,8)	F L .. F R	 code: c6 LFL'BLF'L'B' (8,8)	R F .. L F
 code: c7 L2FU'F'UFUF2L'FL' (11,13) LU2L2BD'BDB2L2U2L' (11,16)	L R .. F F	 code: c8 LFUF'U'L' (6,6) RUBU'B'R' (6,6)	F F .. L R	 code: c9 R2F2R'B2RF2R'B2R' (9,14)	F B .. B F
 code: c10 B'UFU'BU2RU'R'F' (10,11)	F F .. B B	 code: c11 R2D'RU2R'DRU2R (9,12)	F R .. L F	 code: c12 L2DL'U2LD'L'U2L' (9,12)	L F .. F R
 code: c13	F B F B ..	 code: c14	F F B B ..	 code: c15	L R F F ..

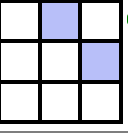
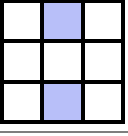
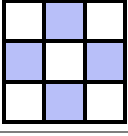
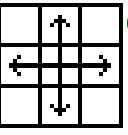
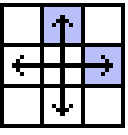
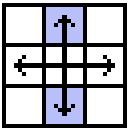
RU2R'U'RUR'U'RU'R' (11,12) L'FURU'F'L2FR'F'L' (11,12) code: c16 F'U'LF'L'F2UFR'F'R (11,12)	L F R F ..	B'UR2D'RU2R'DR2U'B (11,14)	R'FRF'U2R2B'R'BR' (10,12)		
code: c17 BU2B2U'B2U'B2U2B (9,14) LUFU'F'UFU'F'L' (10,10)	F F R L ..	code: c18 F'U'FU'F'UL'ULF (10,10) FUF'UFU'RU'R'F' (10,10)	R L F F ..	code: c19 B'RB'R'B2U2L'BLB' (10,12)	F F B B ..
code: c20 R'FRF'U'F2LFL'UF (11,12)	F B B F ..	code: c21 B'U2BU2BL'B2U'BUL (11,14)	R F F L ..	code: c22 BU2B'U2B'RB2UB'U'R' (11,14)	F L R F ..
code: c23 LU2L'U'LU'L' (7,8) RB2L'B'L B'R' (7,8)	R F .. L F	code: c24 F'U'L'ULU'FU'F'U2F (11,12) BU2B'RB'R'BU'BU'B' (11,12)	F R .. F L	code: c25 B'UFU'BUF' (7,7)	R F .. F L
code: c26 L'BLB'U2B'U2B (8,10)	F R .. L F	code: c27 F'LU2L'FLF'U2FL' (10,12)	F B .. F B	code: c28 FU2F'U2F'LFL' (8,10)	B F .. F B
code: c29 BUB'UBU2B' (7,8) L'B2RBR'BL (7,8)	F L .. F R	code: c30 FURU'R'UF'UFU2F' (11,12) FUF'UF'LFL'FU2F' (11,12)	L F .. R F	code: c31 BU'F'UB'U'F (7,7)	F L .. R F
code: c32	L F .. F R	code: c33	F B .. B F	code: c34	F B .. F B

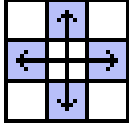
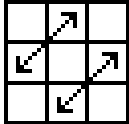
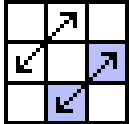
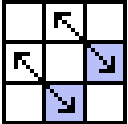
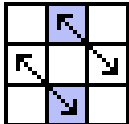
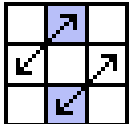
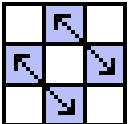
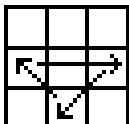
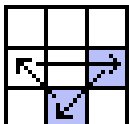
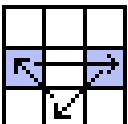
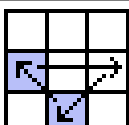
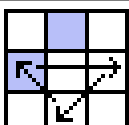
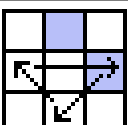
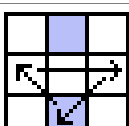
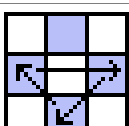
RB'R'BU2BU2B' (8,10)  code: c35 LB'D2BL'U2LB'D2BL' (11,14)	F'U2FU2FR'F'R (8,10)  code: c36 RU2R2FRF'RU2R' (9,12)	FR'U2RF'R'FU2F'R (10,12)  code: c37 BLFL'B'LF'L' (8,8)
 code: c38 RU2RDR'U2RD'R2 (9,12)	 code: c39 F'U2F'D'FU2F'DF2 (9,12)	 code: c40 FR'F'RURU'R' (8,8) L'B'R'BLB'RB (8,8)
 code: c41 F'LF'R2FL'F'R2F2 (9,12)	 code: c42 R2DL'BLD'R2U'F'U'F (11,13)	

四.Edges of Last Layer

一次性还原四个棱块的方向和位置 共 29 个公式。

(下面表格来自 <http://www.speedcubing.com/finallayer.html>)

 code: e1 M2UM'U2MUMUMU2M'UM (13,16,24) RU2R2D'U'F'UFDR2U'R'U' (13,16) R'E'R2E2R'URE2R2ERU' (12,16,22)	 code: e2 MUMUMU2M'UM'UM'U2 (12,14,20) L'RBUL'B'ULR'F'UL'FU' (14,14) FEF2E2FU2F'E2F2E'F'U2 (12,18,24) F2M'UM'U'F2UMUMU2 (11,14,18)	 code: e3 MU2M'U2MUM'U2MU2M'U' (12,16,22) FU2F'UL'U'B'R'U2RU'BUL (14,16) R2U2R'F2U2R2F'R2U2F2RU2R2U' (14,24) F2MF2M2U'M2F2M'F2U (10,16,22)
 code: e4 M2UM2U2M2UM2 (7,12,20) R'L'U2LRFBU2B'F' (10,12) M2DS2D2S2DM2 (7,12,20)	 code: e5 MUM'UMUM'UMUM'U' (12,12,18) L'U2LUFRU2R'U'F' (10,12)	 code: e6 MUM2U2MUMU2MU'M'U'M' (13,16,24) LU'R'UF'L2U'LU'F'R'F2R2 (13,16) FM'FUM2U2M2UF'MF' (11,14,20)

 code: e7		
MU2MU2MUM'U2M'U2M'U' (12,16,22) RBUB'R'F2BD'L'DB'F2 (12,14) MUMFR'F'M2FRF'U' (11,12,16)		
 code: e8	 code: e9	 code: e10
M2U'MU2M2U2MUM2 (9,14,22) LR'D2U2LR'D'L2R2UB2F2 (12,18) RB'R'BFR'B'FR'BRF2U (13,14) M2DS2D'S'M2S (7,10,16)	MU'M2U2M'U'MU2MU'MU'M' (13,16,24) FRUR'U'F2L'U'LUF (11,12)	MUM'UMU2M'U'MU'M'U2 (12,14,20) LUF'U2R'FRFUL'F'UF (13,14) L2SU'S'UL2USUS'U2 (11,14,18)
 code: e11	 code: e11a	 code: e12
MUM'U'MUM'UMU2M' (11,12,18) L2U'LUL'F'L2U'L'UFU'L2 (13,16) M'UMU'F2M'UMU'F2 (10,12,16)	M'U'MUM'U'MU'M'U2M (11,12,18) R2UR'U'RFR2URU'F'UR2 (13,16) M'U'MUF2M'U'MUF2 (10,12,16)	M'U'MU2M'U'M2UM'U2MUM' L'R'F'U2L2U2L2U2L2FLR (12,18) F2MF2U'M2UB2MB2 (9,14,18) F'B'RESRS'R2E'FB (11,12,16)
 code: e13	 code: e14	 code: e15
M2U'MU2M'U'M2 (7,10,16) F2UM'U2MUF2 (7,10,12)	MUM'U2MUM' (7,8,12) F'D2B2L'B2D2F2R'F' (9,14)	M'UMUM'UMU'M'U'MU' (12,12,18) BRUR'FB'DR'D'F' (10,10) SR'FRS'R'F'R (8,8,10)
 code: e16	 code: e17	 code: e18
U'MUMUM'U'MU'MUM (12,12,18) F'D'L'DB'FL'ULB (10,10) LF'L'S'LFL'S (8,8,10)	MU'M2U'M'U2MU'MU (10,12,18) BDL'UL'U'L'UL2U'D'B'U (13,14)	M'U'MUMUMU'MU'M'UM2 (13,14,22) B'U2B2UL'B'D'B2DLBU'B' (13,16) M'U'F2M2U'MUF2UM2 (10,14,20)
 code: e19	 code: e20	
M2UMU'M'U'M'UM'UM'U'M (13,14,22)	M'UMUM'U2M'UM'UMUM'UM' (15,16,24)	

B'U'BRDB2D'B'R'UB2U2B' (13,16) M2UF2UM'U'M2F2U'M (10,14,20)	BLB'F2ULULF'L2U'F2L' (13,16) SUS2U'LULSL'U'L' (11,12,16)	
code: e13a	code: e14a	code: e15a
M2UMU2M'UM2 (7,10,16) F2U'M'U2MU'F2 (7,10,12)	MU'M'U2MU'M' (7,8,12) FRF2D2B2LB2D2F (9,14)	M'U'MU'M'U'MUM'UMU (12,12,18) B'L'U'LF'BD'LDF (10,10) S'LF'L'SLFL' (8,8,10)
code: e16a	code: e17a	code: e18a
UMU'MU'M'UMUMU'M (12,12,18) FDRD'BF'RU'R'B' (10,10) R'FRSR'F'RS' (8,8,10)	MUM2UM'U2MUMU' (10,12,18) U'BDUL2U'LULU'LD'B' (13,14)	M'UMU'MU'MUMUM'U'M2 (13,14,22) BU2B2U'RBDB2D'R'B'UB (13,16) M'UF2M2UMU'F2U'M2 (10,14,20)
code: e19a	code: e20a	
M2U'MUM'UM'U'M'U'M'UM (13,14,22) BUB'L'D'B2DBLU'B2U2B (13,16) M2U'F2U'M'UM2F2UM (10,14,20)	M'U'MU'M'U2M'U'M'U'MU'M'U'M' (15,16,24) B'R'BF2U'R'U'R'FR2UF2R (13,16) S'U'S2UR'U'R'S'RUR (11,12,16)	

注：灰色表示楞块方向错误需要翻转。每种情形的第一种算法只移动 U 与 M 层。

第八节 桥式解法 Gilles Roux Method

根据 Roux 主页他本人提供的方法整理。

一.Left side 1X2X3 block

这一步没有严格的规则，通常来说最好的策略是先建立 1x2x2 的块，再扩展成 1X2X3 的块。

二.Right side 1X2X3 block

此时仍有数千种构型，所以没什么最优方案来还原这五个块。仍然遵照如下原则：

---解决一个 1X2X2 的块

---假如最后一个 C/E 对形成 1X2X3 的块

这一步是为右手设计的。所有动作都可由{U,R,M}完成，有也可借助{F,B}减少步骤，但只用{U,R,M}更具速度竞争优势。

这一步构建 block 时多利用中间层 M 操作简化步骤。即不要完全依赖 F2L 建块。

三.Up coner (CMLL)

首先注意 CMLL 与 COLL 的区别，CMLL 是桥式特有的可以打乱 U 层 M 层的棱。COLL 是层先法的最后一层角还原，仅可以打乱 U 层棱的顺序。

CMLL - solves corners but disturbs U and M slice edges

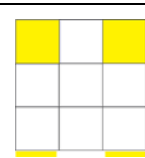
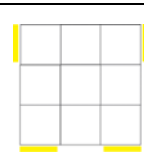
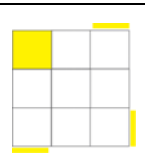
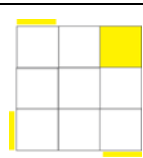
CLL - solves corners but disturbs U slice edges

COLL - solves corners but disturbs U slide edge permutation only (preserves last layer's edges orientation)

OCLL-Oriente the coners, disturbe the edge permutation only.In zz method, U cross has done, OCLL+PLL finish it.

这一步是桥式玩法最公式最多的一步，需要记忆掌握。

1. 先不管四个角的位置，调整它们的顶面颜色朝上。

			
R'U'F'UFR	R'U'F'UFU'F'UFR	R'U'RUR'R'U2R	LUL'ULU2L'

CF'U'RU'RU'LUR' UR'UCF	R'FRUR'FRU'R'F' R	随意用前几个公式	

2. 调整顶角位置

有两种情况：

换邻角： RB'RF2R'BRF2R2U	换对角： FRU'R'U'RUR'F'RUR'U'R'FRF'

如果不分两步，同时调整顶层颜色位置，公式见本节附录。

四. 6E4C

这一步骤需要解决的块有 6 个棱块和 4 个中心块，平均步数大约是 15 步。

1. Edges orientation

判别棱块朝向是否正确

在解决这些剩余的棱块的时候，还是先还原色向，再还原位置。

定义 如果棱块能被 $\{U, D, L, R, F^2, B^2\}$ 生成的子群还原，那么它的色向是正确的。

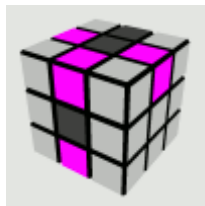
具体的判断方法就是：

- 1). 对于 UL 和 UR 棱，如果它的 L/R 面处于魔方的 F/B/L/R 面上时，它的色向是正确的；
- 2). 对于其它的四个棱块，如果它的某个面和“与此面同色或相对色的中心块”相邻，那么它的色向是正确的。

可以根据上述定义推导出更为简捷的判断方法是，U 或 D 面中心块始终放在 U/D 面，看楞块的侧面贴片即楞块的 L/F/R/B 贴片，如果出现在 U 或者 D 面上，这一块的朝向就是错误的。

这个判断本身也提供了翻转楞块朝向的方法。M 层上的楞块在 M^2 , M'^2 转动下，朝向的正误不变。只通过一次 M, M' 转动，再通过 U 转动转离 M 层，那么它的朝向正误翻转。为了保证 UD 中心块始终处于 U/D 面上，每次操作必须包含偶数次 M 转动，如 MUM, MUM' 都是合规的，UMU 则不合规。那么那些经历偶数次 M 动作的楞块朝向正误不变，经过奇次的，正确

朝向变为错误，错误的变为正确。所以，翻转朝向的方法必须如此精确设计，使得错误的棱经历奇数次 M 动作，正确的棱经历偶数次 M 动作。



例如对这个种情形，操作 $M' U M'$ 即将所有棱朝向更改为正确。这个操作过程中，F 面上两个错误楞块经历一次 M' ，B 面上的两个正确楞块经历两次 M' ，顶层两个错误块经历一次 M' 。所有六个楞块最后都变为朝向正确的。

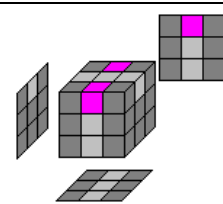
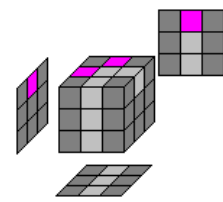
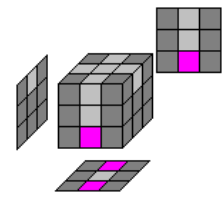
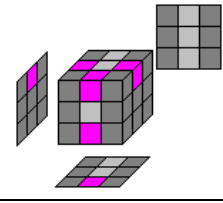
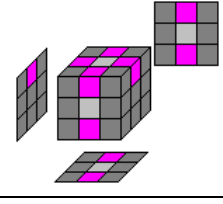
另外，朝向错误的楞块只能是偶数块。即 0，2，4，6 块。而且仅有 4 可以直接 $4 \rightarrow 0$ 。要想把这个数字削为零，必须经过上面的标准步骤，即 $XX \rightarrow 4 \rightarrow 0$ 。

到了这里有必要重新理解一下楞块朝向的对错这个概念。所谓正误就是在特定的转动子群下能否归位。显然，如果允许所有的转动操作，任何楞块最终都能还原，也即朝向是正确。本例中，后续步骤只能用 $\{U, M2\}$ 时，楞块的贴片在魔方侧面的不可能通过 $\{U, M2\}$ 生成的操作变到魔方的顶面或底面，必须在这之前利用 M 操作把它们的侧面贴片放到侧面。

还原棱块朝向

在知道上述棱块色向的规则之后，首先判断哪几个棱块色向不对，下图中的紫色块表示色向错误的棱块分布。使用对应公式直到所有的棱块的色向正确为止。学会技巧之后这些公式即可忘掉。

E1		[E1.] $M' U2 M \ U2 y2 \rightarrow E8$
E2		[E2.] $M U2 M \ U2 y2 \rightarrow E8.$
E3		[E3.] $M2 \ U \rightarrow E8.$
E4		[E4.] $M' U' M' \ U' y2 \rightarrow E8.$

E5		[E5.] MUM' U'y2 --> E8.
E6		[E6.] M'UM U2 --> E8.
E7		[E7.] M'U'M' U --> E8.
E8		[E8.] M'U'M' --> Job's done
E9		[E9.] M'U'M' U2y2 --> E4.

2.Finish L/R-sides

还原顶层 UL 棱块和 UR 棱块。有一个限制就是，不能破坏步骤 4a 中还原好的色向。为了不破坏，U 和 M2 操作是没有问题的，但如果想转 M(M')，那么在它后面必须跟上一个 U2M(U2M') 以便复原色向。即总是使 M 层上的四个棱经历偶数次 M 动作，它们的朝向始终保持正确，U2 则保持它们在 M 层。情况很少，下面是几个例子（R-白，L-黄）：

		
U'MU2M	M2UMU2M'	U2M'U2M'UM2

注意，UL/UR 两个棱不可能先后各自单独插入 RL 面，必须要把 UL/UR 调到 M 层上，放在相邻的的两个棱上，才能同时正确插入。让插入的大块里各个小块相对位置正确，作为一个整体插入，这是插入技巧的全部。

3.Permute M-edges

复原余下几个楞块位置。这一步很简单，只用三个公式：U2MU2, U2M2U2 and E2ME2，即可将剩余棱块归位。

小结第四步，第四步是在完全不动左右两个 block 情况下完成的，调正棱朝向我们被限制用 {U, M} 生成动作。调整 UL/UR 棱时要保持这 6 个棱朝向不动，我们进一步被限制到 {U, M2, M+U2M}。最后调整 M 层棱时，进一步要保持 UL/UR 棱不动，只有 {U2, M2, U2M+U2M} 可用。这种逐步限制的特点体现了桥式的思想，后一步的每一个动作尽量不要移动前一步的结果，而不仅仅是后一步的结果不改变前一步的结果，逐步缩小操作转动群削减状态数最终还原。

速度解法

CMLL 一步还原顶角朝向和位置

Orientation:

A	B	C	D	E	F	G	H

Permutation

1	2	3	4	5	6

One step CMLL:

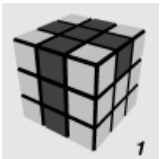
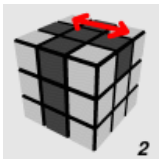
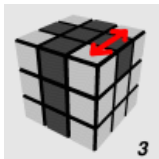
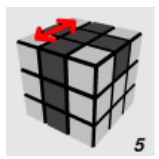
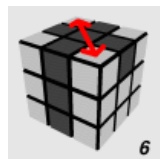
A2 (A2') (R'U)L'U2(RU'R')U2R2 RB'RF2R'BRF2R2 [U']F'LF'R2FL'F'R2F2 [U']RU'L'UR'U2LU'L'U2L	A6 (A6') (R2U'RF'R'U)(r2U)(FUF') R2DL'BLD'R2U'F'U'F R'UR'U'B'DB'D'B2R'B'RBR	B1 (C1') (R'U'RU')(R'U2R) [U](RU')U'R'U'(RU')R' [U']LU2L'U'LU'L' [U']RB2L'B'L B'R'
B2 (C4') (I'URU')(rDL'B'LD')(r'U) [U2](R'FU2)(F'RF'R')(U2RF') [U']RUR'URU2B'R'BL'B'L R'FRF'LUL'F'LU'L'F L'BU2B'LBL'U2LB' [U']F'LU2L'FLF'U2FL'	B3 (C2') (B'RBR')U2(R'U2R) LR'U'RUL'U2(R'U)U'R [U']L'BLB'U2B'U2B [U]R'FRF'U2F'U2F	B4 (C3') (LU2L')U2(R'U)(LU') [U2](RU')U'(R'U)U(R'F)(RF') [U']FU2F'U2F'LFL'
B5 (C5') (R'U)(LU')(RUR') [U']B'UFU'BUF' [U2](R'F)(RF')r(UR')	B6 (C6') (RU2R')(FR'F'R)(U'RU'R') (RU2)(I'UR'U')(IU'RU')R' [U']F'U'L'ULU'FU'F'U2F [U']BU2B'RB'R'BU'BU'B'	C1 (B1') (RUR'URU2R') (z')(ULU'LUL2U) [U']BUB'UBU2B' [U']L'B2RBR'BL

 C2 (B3') (zU')(R2UR'I')y(RU'R'U) (L'U2)(LU2)(LF'L'F) [U2](R'U)URUF(R'U)(RU')F' [U']F'U2FU2FR'F'R	 C3 (B4') (FR'F'R)U2(RU2R') [U]B'U'R(UR')BU(RU')U'R' [U']RB'R'BU2BU2B'	 C4 (B2') (IU'R'U)x2(r'U'rD)(r'Ur)D' [U]R'U'RU'R'U2FR'FL'F' RB'R'BL'U'LBL'ULB' LF'U2FL'F'LU2L'F [U'](FR')U2(RF')(R'F)U2(F'R)
 C5 (B5') (RU')(L'UR'U'R) [U']BU'F'UB'U'F (RU')(r'F)R'(F'r) [U']BU'F'UB'U'F	 C6 (B6') (R'U2)(IU')(RUI')U(R'UR) [U2]R(UR'U)(R'F)(RF')RU2R' [U']FURU'R'UF'UFU2F' [U']FUF'UF'LFL'FU2F'	 D1 (D1') (z'U')(RU'R)U'(LU)(R'U)(R'U) (R'D)(R'D)(R'U)(RD')(RD'R) [U]R(UR'U)(RU')(R'U)(RU')(R'U) (RU')U'R' [U']LB'D2BL'U2LB'D2BL'
 D2 (E5') z(U'R2z)(R'U'R)D2(R'UI2) (R'F2)(R'U'R)F2(R'UR2) L'U2L'D'LU2L'DL2 BUL'BLB2U'L'B'L [U2](R'U)(UR')D'(RU')U'(R'D)R2 [U']F'U2F'D'FU2F'DF2	 D3 (E3') (L'U2L)U(fU'f'UfUf') (yzU)R2z(RUR')D2(RU'R2) FU2FDF'U2FD'F2 [U']B'U'RB'R'B2URBR' [U'](RU')U'RD(R'U)U(RD')R2	 D4 (F5') (LF'L'F)(UFU'F') [U'](r'U')(R'U)(LU')(RU) [U']R'U'r'UBU'B' [U']FR'F'RURU'R' [U']L'B'R'BLB'RB [U]R'F'L'FRF'LF RU'r'U'F'UF
 D5 (F3') (F'LFL')(U'L'UL) (IU'LU)(R'U)(L'U') [U]R'U'r'UBU'B' F'r(UR')U'(r'F)R [U']BLFL'B'LF'L'	 D6 (D6') (RUR')(L'U2)(RUR')(U2L) RUL'R'U2RUR'U2R FU2F2LFL'FU2F' [U](L'U)U(RU')(R'U)UL(RU')R' [U']RU2R2FRF'RU2R' RUR2F2rFR'F2R	 E1 (F1') (R'U'RU')(R'U2)(R2UR'U)(RU2R')) (R2F')(UFU'F')(U'F2)(RF'R) RB'R'BU'FU'BUF'UB' L2FU'F'UFUF2L'FL' LU2L2BD'BDB2L2U2L'
 E2 (E2') (U'y')(F'RUR')(U2y')(L'UR'U'L) [U](RU'L'UR')U2(B'UBL) BU'F'UB'U2L'ULF [U']R'U'RF(R2D')R(UR'D)(R2U') F' B'UFU'BU2RU'R'F'	 E3 (D3') (RU')(r'U'r'U'r'U')y'(R'U2R) L2DL'U2LD'L'U2L' [U]RB'R'U'B2RBR'UB [U2]R2D(R'U)U(RD')(R'U)(UR')	 E4 (F4') (IU')F(R'U'RF'R'U) (R'U)U(RF')(R'F)U2(F'R)F RB'UR'B'RU'R'B
 E5 (D2') (R'FRUR'FRU)y'(RU2R') (R2D'R)U2(R'DR)(U2R) (I'UIUI'UI'U)y'(RU2R')	 E6 (E6') [U'](FRUR'U'F') LFUF'U'L' R'U2RFU'R'U'RUF'	 F1 (E1') (RU2R'U'RU')R2U2(RU)(R'UR) (FR')(FR2)(U'R'U')(RUR'F2) BU'FU'B'UF'UB'RBR'

[U']L'BLUB2L'B'LU'B' [U]F2D'FU2F'DFU2F [U]F'U2FLU'F'U'FUL'	RUBU'B'R'	RU2R2F2D'F'DF'R2U2R' [U]LUL'ULU2L'R'U'RU'R'U2R
F2  (F2') (R'U)r(U2R'x)(R'URU'r) R'UrU2R2FRF'R (R'U)(RU')U'r'(R'F)(RF')R R'URU2R'L'URU'L FRF'LFR2FRF2L'	F3  (D5') (F'U'FU)(FR'F'R) [U'](R'U'RU)(IU'R'U) [U']BUB'U'r'U'R [U'](FRF')r(UR')U'R' LFL'BLF'L'B' [U'](L'B'RB)(LB')(R'B) [U]R'F'LFRF'L'F	F4  (E4') F(R'U'RF'R'U)(F'R) FU'(r'F)R2F'rUR2F' F'L2F'R2FL2F'R2F2
F5  (D4') [U'](RUR'U')(I'URU') (FUF'U')(F'LFL') [U'](F'U'FU)(rUr') [U']R(UR')U'(R'F)(RF') R'F'RB'R'FRB	F6  (F6') (I'URU')(IU2R'U'y')(R'U'R) (R'FRF')(RU2R')(U'F'U'F) (F'UFU)(RU2R')(FR'F'R) F2R2FU'L'ULF'R2F2 (R'U)R2Dr'U2Rd'(R2U'R) B2L2BR'U'RUB'L2B2	G1  (G1') (FRUR'U'RUR'U'F') (BULU' L'ULU'L'B') R'U2R2UR2UR2U2R' (RU')U'(R2U')(R2U')(R2U')U'R [U']BU2B2U'B2U'B2U2B [U']LUFU'F'UFU'F'L'
G2  (G4') (R'U2)(RUR'U'RU2)(IU'R'U) R'F2RF'U2RU'r'U'F [U](RU')U'R'U'(RU')R2U(LU')R(UL') [U']B'U2BU2BL'B2U'BL	G3  (H2') [U'](R'FRU)(FU'RUR'U'F') RB2U'B2UB2UB2R' [U]F(R2U')R2UR2UF'y(RU')U'R' [U']R'FRF'U'F2LFL'UF	G4  (G2') (RU2R'U'RU)(R'U2I')(URU') RB2R'BU2R'UrUB' [U'](R'U)UR(UR'U)(R2U')(r'F)R'(F'R) [U']BU2B'U2B'RB2UB'U'R'
G5  (H3') (RU')(L'U)(R'U)(LUL'UL) R'FR'F'R2U2B'RBR' (RU')(r'F)(R'F)RU(r'F)R [U']B'RB'R'B2U2L'BLB'	G6  (G6') (U'y')(R'U'RU')(R'U)y'(R'UIU) L'U'LU'L'UB'UBL (RF')U'(R2F)U'F'UR2U(FR') [U'] F'U'FU'F'UL'ULF [U'] FUF'UFU'RU'R'F' [U2]R'U'RU'R'UF'UFR	H1  (H1') (LUL'ULU'L'ULU2L') [U](R'FRUR'FRU'R'F'R) BU2B'U2B'U2B'U2B [U](RU')U'R'U'R(UR')U'(RU')R' [U]L'FURU'F'L2FR'F'L' [U]R'FRUR'FRU'R'F'R RUR'URU'R'URU2R'
H2  (G3') F(URU'R'U)(F'U')(R'F'R) [U']R(B2U')(B2U')(B2U)B2R' [U](RB')(R'B)U2R2F'(RU')r'F2R2 [U'] F'U'LF'L'F2UFR'F'R	H3  (G5') (RU)(R'U)(RUL'UR'U'I) (FR'F'R)U2F2(LFL'F) [U2]R(UR'U)RU(r'F)R'(F'R) [U]R'FRF'U2R2B'R'BR' [U2]LUL'ULUR'UL'U'R	H6  (H6') [U'](FRUR'U'RUR'U'RUR'U'F') (IU)(FU'F'UFU'F'UFU'F'I') RBUB'U'BUB'U'BUB'U'R' R'UF2D'FU2F'DF2U'R RUR2F'RF'U'FL'U'LF [U']R(UR'U)y'(R'U)(RU')(R2F)(R

		F'R) [U'] B'UR2D'RU2R'DR2U'B R2B'D'R2E'F2R2URF2
A2 = A3 = A4 = A5, H2 = H4, H3 = H5.		

Roux 本人的公式如下(此表格中 **M,E,S** 与本文约定 **MES** 方向相反,其余一致):

Analyze the orientation pattern and find the right column						
	I (A1)' Sym(A1)	R'UL'U2RU'R'U 2R2 (A2)' (A3)' (A4)' (A5)' Sym(A2, A3, A4, A5)	(A2)' (A3)' (A4)' (A5)' Sym(A2, A3, A4, A5)	(A2)' (A3)' (A4)' (A5)' Sym(A2, A3, A4, A5)	(A2)' (A3)' (A4)' (A5)' Sym(A2, A3, A4, A5)	R2U'RF'R'Ur2U FUF' (A6)' Sym(A6)
	R'U'RU'R'U2R (C1)' Sym(C1)	R'FU2F'RFR'U2 RF' (C4)' Sym(C4)	R'FRF'U2F'U2F (C2)' Sym(C3)	 (C3)' Sym(C2)	R'ULU'RUR' (C5)' Sym(C5)	RU2R'U'RU'BU B'U'R' (C6)' Sym(C6)
	 (B1)' Sym(B1)	 (B3)' Sym(B4)	 (B4)' Sym(B3)	 (B2)' Sym(B2)	 (B5)' Sym(B5)	 (B6)' Sym(B6)
	(R'DR'DR')U(R' DR'DR')' (D1)' Sym(D1)	R'F2R'U'RF2R' UR2 (E5)' Sym(D3)	 (E3)' Sym(D2)	RU'r'U'F'UF (F5)' Sym(D5)	 (F3)' Sym(D4)	RUR2F2rFR'F2 R (D6)' Sym(D6)

	R2F'UFU'F'U'F2 RF'R (F1)' Sym(E1)	RU'L'UR'U2B'U BL (E2)' Sym(E2)	 (D3)' Sym(E5)	RB'UR'B'RU'R' B (F4)' Sym(E4)	 (D2)' Sym(E3)	FRUR'U'F' (E6)' Sym(E6)
	 (E1)' Sym(F1)	R'UrU2R2FRF' R (F2)' Sym(F2)	 (D5)' Sym(F5)	 (E4)' Sym(F4)	 (D4)' Sym(F3)	R'URUBU2B'RB 'R'B (F6)' Sym(F6)
	 (E6)2 (G1)' Sym(G1)	R'F2RF'U2RU'L 'B'U (G4)' Sym(G4)	R'FRUFU'RUR' U'F' (H2) ' (H4)' Sym(G3)	 (G2)' Sym(G2)	RU'L'UR'ULUL' UL (H3) ' (H5)' Sym(G5)	R'U'RU'R'UF'UF R (G6)' Sym(G6)
	(R'FRUR')F(R'F RUR')' (H1)' Sym(H1)	 (G3)' Sym(H2, H4)	 (G5)' Sym(H3, H5)	 (G3)' Sym(H2, H4)	 (G5)' Sym(H3, H5)	(E6)3 R2B'D'R2E'F2R 2URF2 (H6)' Sym(H6)

第九节 Petrus Method

根据 Lars Petrus 网页他本人提供的方法整理。常用于最少步数玩法。分七步：

1. 2x2x2 块
2. 扩展成 2x2x3 块
3. 修正朝向错误的棱块 (Edge Control)
4. F2L-3rd & 4th (还原剩下的两对 F2L, 步骤 c 将使结果的顶层 Cross 已完成)
5. 顶层角块位置
6. 顶层角块方向
7. 最后 4 个棱块的位置

此节中，我们以上黄下白，左蓝右绿，前红后橙为例。在左下侧建这个 2X2X3 block，留着 U 面和 R 面可以自由转动。最后只用 U, R 转动复原剩下的块，这是这个方法的核心特点。从理论上讲，这个特点会保证较少的步数还原魔方。

Petrus 方法有很大的灵活性，可以有简单玩法，重复步数多，公式少，简单好记。也有高级玩法，第三步调整楞块朝向一步完成，第四步两对 F2L 一步到位，第五六七步适当合并。这使得它可以作为速度玩法和最小步玩法。

Step 1. Build a 2x2x2 block

做一个 2x2x2 的块。或者等价的说，还原一个角块以及跟它相邻的三个楞块。

Step 2 - Expand to 2x2x3

将 2x2x2 扩展成 2x2x3 的块。这个过程中不要移动上一步建好的 2x2x2 块。作为进阶，可以考虑用打破 2x2x2 块的更少步数公式。

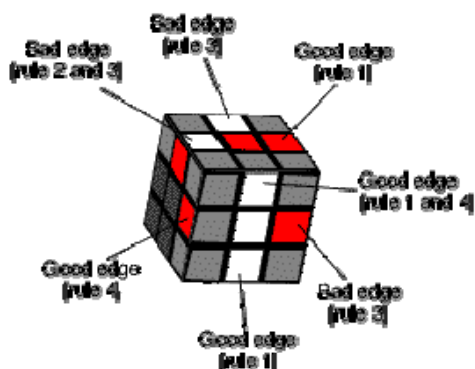
Step 3 - Twist the edges --- 2, 4 or 6 bad edges

楞块朝向的好坏定义同 ZZ, Thistlethwaite, Roux 等方法里面的判断。但对于 Petrus 方法有更简洁的形式。

定义：能被 <U,R> 转动还原的楞块，它的朝向是好的。反之，是坏的。

Here are the four rules to remember:

1. An edge in the A layer that has an A color sticker next to the center is good.
2. An edge in the A layer that has an A color sticker away from the center is bad.
3. An edge in the B layer that has an A color sticker next to the center is bad.
4. An edge in the B layer that has an A color sticker away from the center is good.



将坏的朝向调整为好的，分三种情况：

1. Two bad edges
2. Four bad edges
3. Six bad edges

初级玩法只有一个公式，**L'U'L**，一次同时翻转两个棱的朝向（图中两个白色的棱）：



重复这一步骤即可翻转所有棱朝向。为了竞速，针对 21 种错误状态有 21 个公式，这里不一一列举了。

这一步保证了第四步完成后顶层自动形成十字，减少了顶层状态数，所以在第六第七步处理顶层比 CFOP 所需公式要少。

Step 4 - Finish 2 layers

这一步只转动 U, R 两个活面，左下角 2X2X3 block 不可动，将 R 面的两个 F2L 补齐，变成 2X3X3 block。第三步调整楞块朝向，第四步仅用 UR 转动，这是 Petrus 方法与 Fridrich 区别之所在。

这一步分两小步：

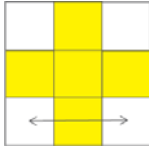
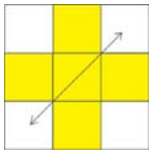
1. 将 R 面的前下角块和与它相邻的两个棱块同时还原，或者将 R 面的后下角块和与它相邻的两个棱块同时还原。即先还原一个 1X2X2 的方块。
2. 还原剩下的两个 F2L。

Step 5 - Position the corners

到了这一步，魔方只剩下最后一层要处理。而最后一层楞块朝向已经调整好，它的状态数要少于 CFOP 最后两步所面临的。和 ZZ, ZB, VH 方法的顶层情况完全一样，处理方法也可以

一样。可以 **OCLL+PLL**，或者 **COLL+EPLL**，或者 **ZBLL** 一步完成，当然，如果第四步利用 UD 以外的转动把顶层楞块的 Phase 也调整好，还可以用 **ZZLL** 一步完成最后一层。这正是 Petrus 提供的高级玩法。他提供的初级玩法是顶层分三步完成：**PCLL+OCLL+EPLL**。只需要用三个公式：**Niklas**，**Sune**，**Allan**。

第五步目标是还原顶层四个角块位置，其颜色朝向留第六步扭转。有 $1/6$ 的几率四个角块位置都正确，直接进入下一步，剩下两种情况需要互换两个角块，只需要一个公式 **Niklas=BU'F'UB'U'FU2**：

	4/6 的几率，互换相邻角块 一次 Niklas=BU'F'UB'U'FU2
	1/6 的几率，互换对角角块 两次 Niklas 转动，第一次把对角变换为邻角，第二次同上。

Step 6 - Twist the corners

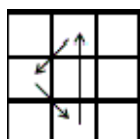
这一步扭转 U 层角块使其方向正确。只有一个公式 **Sune**。 **Sune=(RUR')U (RU2R')**。 **Sune** 也即 **OLL27**。

	Case 1, 有一角朝向正确，转动使其处于右前位，使用一次 Sune ，所有角块朝向即还原。
	Case 2, 无角块朝向正确，转动使某个顶角块处于右前位，并且它的 U 面处于魔方的 F 面。使用一次 Sune ，变为 Case 1.
	Case 3, 两角块朝向正确，转动使某个错误朝向顶角处于右前位，并且它的 U 面处于魔方的 R 面。使用一次 Sune ，变为 Case 1.

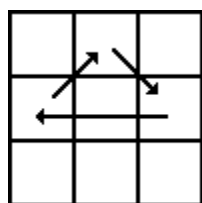
显然，这一步可以直接用 **OLL21-27**。

Step 7 - Position the edges

只剩下四个楞块的位置需要调整。可以重复利用一个公式 **Allan=L2U'BF'L2B'FU'L2**。它就是 PLL 的 U Permutation-a 转动就 90 度的情形：

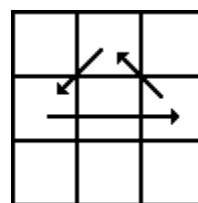


可直接利用 PLL 的子集 EPLL 完成这一步。



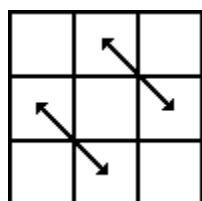
U Permutation : b

$(R' U R') U' R' U' (R' U) R U R^2$



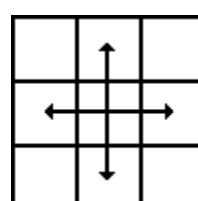
U Permutation : a

$(R^2 U') R' U' R U R U (R U' R)$



Z Permutation

$R' U' (R U') R U (R U') (R' U) R U (R^2 U') R'$



H Permutation

$r^2 (R^2 U) r^2 (R^2 U) U r^2 (R^2 U) R^2 r^2$

第十节 Heise method

由 Ryan Heise 创造，不需要算法，是一种有一定难度的方法。此节根据 Heise 网页其本人提供的方法整理。

一. 造四个 1X2X2 方块

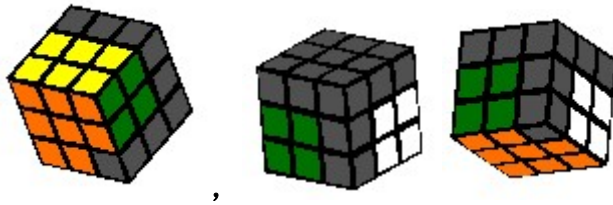
二. 合并四个 1x2x2 块并调整剩余棱块方向

完成此步后整个魔方只剩下顶层和一个 F2L 未还原，棱块色向已调整好

三. 还原剩下的 5 个棱块和任意 2 个角块

四. 还原最后 3 个角块

Step 1. build the structure like this:



Heise 方法的最初形式是基于 Petrus 思路，其第一步是要完成上图结构。

转动到第三个图，橙色为底，绿色为前，白色为右。左下侧是一个 2X2X3 的大块，这个大块就是 Petrus 前两步完成的结构。Petrus 方法其后完成 F2L，Heise 方法与之区别之处在于第一步就完成 F2L 的 2/3，即建立了右侧那个白色方块，而剩余的一对 C/E 块又分别在两步中先后完成 E 块 C 块。

Step 2 完成中间层楞块和顶层十字，顶层十字只要求方向正确，楞块位置第三步调整



这一步可以不用算法自己调整好。为了达到最少步数，可以用算法（从略）。

Step 3. 还原底角同时调好顶层棱块位置



这一步有 17 个算法。

Here is Ryan's page including helpful tips for seeing the permutation of the edges in step 3.

1). Corner in Place

 0/10	 0/7	 7/7
$x(R' U R')x'D2x'(L U' L')xD2x(R2 U')$	$((R U)(R' U R))(U2 R')$	$((R' U')(R U' R'))(U2 R)$
 0/11	 0/11	
$x(U L2 U')x'(U R2 U')x(U L2 U')x'(U R2)$	$x'(R' U2 R)xy'(U' R2 U)yx'(R' U2 R)x(d' L2)$	
 0/9	 0/9	
$(R2 U2)((R' U')(R U' R'))(U2 R')$	$(R2 U2)((R U)(R' U R))(U2 R)$	
 0/11	 0/11	
$((R U')(R' U' R)(U R'))U2(R U' R')$	$((R' U)(R U' R' U'))(R U2)(R' U R)$	

2). Corner on Top

 0/7	 0/7
$(L' U R2 U')(L U R2)$	$(R2 U2 R2 U' R2 U' R2)$

	
0/7	0/7
((R2 U)(R2 U))(R2 U2 R2)	
	
0/9	0/11
(U' L)y'(R2 U R2 U2 R2 U R2 d R')	
((R U')(R' U))(R U R' U)(R U' R')	

3). Corner on Top, Twisted

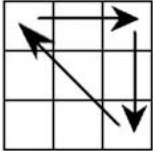
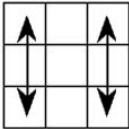
You will notice that the last F2L corner to be placed will start out directly above its final position in the LL. Once you see the case you have, you do U, U', or U2 and then do the sequence. After some practice, you won't need to place the corner over its final position and do a U move.

			
0/7	0/7	0/7	0/7
(U')y'(R D)(R' U R)(D' R') (U)(R' D')(R U' R')(D R) (U')(R U' R' U2)(R U' R') (U)(R' U R U2)(R' U R)			
			
0/7	0/7	0/7	0/7
(U')((R U' R' U' R)(U2 R')) (U)((R' U R U' R')(U2 R)) (d)((R' U2)(R U')(R' U' R)) (U')(R U2)(R' U R U' R')			
			
0/7	0/9	0/9	0/9
(U2)(l U)x y U2(R' U' R U')(l' U) (d2)x'(U' l')U2(R U R' U) (l U) x(R' U R)x'U2x(L U L')x'y'(U2 R') x(L U' L')x'U2x(R' U' R)x'y'(U2 R)			

Step 4. 还原四个顶角

这一步调整好四个顶角位置和朝向。有 42 个算法：

0 corners twisted

	
$x(R' U R')D2(R U' R')(D2 R2)$	$x'(R U' R)D2(R' U R)(D2 R2)$
	
$x'(R U' R' D)(R U R' u2)(R' U R D)(R U R')$	$(r2 R2 U)(r2 R2 U2)(r2 R2 U)(r2 R2)$

1 corners twisted

2 corners twisted

3 corners twisted

剩余公式待总结

这一步全部 41 个公式，Bernard Helmstetter 给出：

http://www.ai.univ-paris8.fr/~bh/cube/solutions_c2.html

第十一节 ZZ method

ZB method 由 Ron van Bruchem 和 Zbigniew Zborowski 各自独立在 2003 年发展起来。有数百个公式。此处只简介其基本步骤：

底层十字；

三对 F2L；

完成最后一对 F2L 同时架好顶层十字（ZBF2L）；

顶层还原（ZBLL）（角块还原与楞块位置还原）。

VH method 由 Lars Vandenberg 和 Dan Harris 创造，做为 Fridrich 方法向 ZB 方法的过渡。公式很多。

底层十字；

三对 F2L；

完成最后一对 F2L 同时架好顶层十字；（VHF2L）

完成顶层角块的 OLL 同时还原角块位置；

顶层楞块的 PLL。

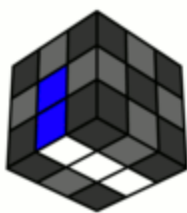
下面主要来介绍 ZZ method. This method was created in 2006 by Zbigniew Zborowski, the co-creator of the ZB method. It has three basic steps: EOLine, F2L, and LL.

The Steps

- **EOLine:** This is the most distinctive part of the ZZ method. In this step, the solver orients all the edges while placing the DF and DB edges. The two edges and the bottom centre are the "line" in EOLine. This step puts the cube into an <L, U, R> group, meaning F, B, or D moves are not required for the remainder of the solve. Although this step may seem like a hinderance, it speeds up the F2L and LL.
- **F2L:** The solver creates a 2x3x1 block on each side of the line via blockbuilding. Because one only needs to do L, U, and R moves, solving is very quick.
- **LL:** The solver uses algorithms to solve the remaining pieces. Since the edges in the LL were oriented during EOLine, it can be completed in fewer moves and/or with fewer algorithms to learn.

一. EOLine

这一步扭转所有的楞块到正确方向。平均 6.127 步。通常分为两小步：先调整棱向，再底面一字（通常是 DF，DB 两块）。即 E0+Line。



定义：如果一个楞块可以通过 R, L, U, D（或 RLUDF2B2）操作还原，就称它朝向正确。



上图是一个例子，左图所有楞块朝向都是正确的。右图所有楞块朝向都是错误的。

根据上面定义可知，F 或 B 操作使得 F 或 B 面上朝向正确的楞块扭转为错误，朝向错误的楞块扭转为正确。F2, B2 则不改变正误。所以，F/B 操作是扭转色相的核心技巧。

E0 不需要记公式。因为只能是 0, 2, 4, 8, 10, 12 个错向楞块，按下面两个步骤来即可全部翻转正确：

≥4 个错向楞块，RULD 操作，移动四个到 FB 面，动作 F/B 即把这四个错向翻转正确。

2 个错向楞块，只将一个移动至 F/B 面，F/B 即变成前一种情况。

熟练之后即可将 E0 和 Line 合并，一步做到 E0Line。

二. F2L - Completing the First Two Layers

This stage completes the first two layers by building two 1x2x3 blocks on either side of the Line made in the previous stage.

为了简单，通常分步做。先做 1X2X2 的块，再把剩余 1X1X2 块加上去，形成一个 1x2x3 的块。

1x1x2 Blocks

只能用 ULR 几个动作。下面是三个基本状态：

		
$RUR'RUR'URUR'RUR'RUR'RUR'$	$RU2R'RU2RUR'$	$RU2R'URUR'RUR'RUR'RU2RUR'$

为了提速 1X2X2 块构建要掌握一定技巧与公式。熟练到一定程度可以依据情况交叉处理左右的块。

三. LL - The Last Layer

第二步结束顶层棱的 U 面自动朝上，但相对位置需要调整。第三步要做的是顶层棱块对位，顶层角块对色对位这三件事，三件事的先后顺序不同，对应不同的方法。

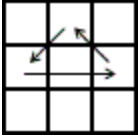
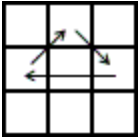
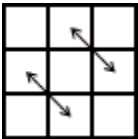
1. OCLL+PLL，即先将角块对色（OCLL），然后同时将顶层角块楞块对位（PLL）。与 CFOP 的 OLL+PLL 很相似，优势在于 OCLL 的算法要少。OCLL 最小需要 6 个算法，PLL 最小需要 14。OCLL Algorithm:

R U2' R2' U' R2 U' R2' U2 R	(y) R U2 R' U' R U R' U' R U' R' R U R' U R U' R' U R U2' R' (y) F R U R' U' R U R' U' R U R' U'	R U R' U R U2' R' (y') R' U2 R U R' U R (y2) L U L' U L' U2' L' (y) L' U2' L U L' U L	R U2 R' U' R U' R' (y') R' U' R U' R' U2 R (y2) L U2' L' U' L U' L' (y) L' U' L U' L' U2' L
R2' D R' U2 R D' R' U2 R' (y2) R' U' R U' R' U2 R2 U R' U R U2' R'	I' U' L U R U' r' F (y') R U2 R' U' R U' R2 U2 R U R' U R	I' U' L' U R U' L U (x') (y2) R' F R B' R' F' R B	

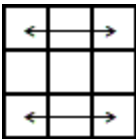
PLL

Permutations of Edges Only:

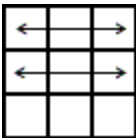
	M2 U' M2 U2' M2 U' M2 R2' r2 U' L2 I2' U2 R2' r2 U' L2 I2' U' M2 U2 M2 U M2 U2 M2 F2 M2 F2 U' F2 M2 F2 U x U2 M2 U2 B' U2 M2 U2 B R2 U2 R2 U2 R2 U R2 U2 R2 U2 R2 U' L R U2 L' R' F' B' U2 F B R U2 R' U' R' U' R2 U' R2 U2 R2 U2 R' U R2 U2 R U2 R2 U2 R2 U2 R U2 R2 M2 U M2 U M2 U M2 U M2 U M2 U' M2 U M2 U2 M2 U M2
--	---

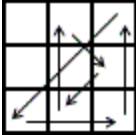
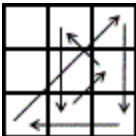
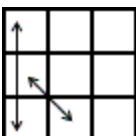
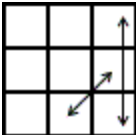
	R' D' R D' B D B' D R' D' R U R' D R D' B D' B' D R' D R U
	(y2) F2 U' L R' F2 L' R U' F2 M2 U M' U2 M U M2 (y2) M2 U M U2 M' U M2 R2 U' R' U' R U R U R U' R (y2) L2 U' L' U' L U L U L U' L B2 U' M' U2 M U' B2 (y2) R U' R U R U R U' R' U' R2 M2 U M' U2 M U2 M' U2 M U M2 (y) R2 U' y r U2 r' R U2 R' y' U' R2 R U R' U' L' U' L U2 R U' R' U' L' U L
	(y2) F2 U R' L F2 R L' U F2 M2 U' M' U2 M U' M2 (y2) M2 U' M U2 M' U' M2 R' U R' U' R' U' R' U R U R2 (y2) R2' U R U R' U' R' U' R' U R' B2 U M' U2 M U B2 L2 U L U L' U' L' U' L' U L' R U' R U R U R' U' R' U' R' U2 R' M2 U' M' U2 M U2 M' U2 M U' M2 L' U' L U R U R' U2 L' U L U R U' R'
	y x' R U' R' U D R' D U' R' U R D2' F x y M2 U M2 U M' U2 M2 U2 M' U2 y M2' U' M' U2' M2' U2' M' U M2' U R' U' R U' R U R U' R' U R U R2 U' R' U R2 U R2 U' R2 F2 R2 U' F2 U R2 F2 y' M2 U2 M' U' M2 U' M2 U' M' U R' U L' U2 D2 R U' L R' U L' U2 D2 R U' L R' U L' E2 L U' R L' U R' E2 R U' L' y R2 U' R2 U R2 x' U2 R2 F U2 F' R2 U2 M' U M2 U M2 U M' U2 M2 U' U' I' U R U' D' R U D' R U' R' D2 d R U R2 U' R' F R U R U' R U' R' U' R U R' F' M2 U' M E2 M E2 U M2 F2 M2 F2 M2 U M2 U M2 U2 U2' R U R U' R U2 R U' R' U' R U R' U' R' U R U' R U' R U' R U' R U' R U' R y R U R' U R' U' R' U R U' R' U' R2 U R U2 y' R U R B' R' B U' R' f R U R' U' f' (x2 y) M2 U S2 U' S' M2 S M2 U (F2 M2 F2 M2) U' M2

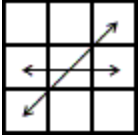
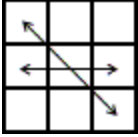
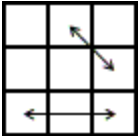
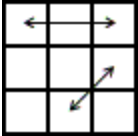
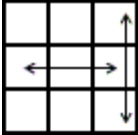
Permutations of Corners Only

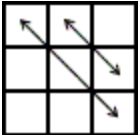
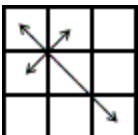
	$R' F R' B2 R F' R' B2 R2$ $(x') R' D R' U2 R D' R' U2 R2$ $(x) R' U R' D2 R U' R' D2 R2$ $(y' x) L2 D2 L' U' L D2 L' U L'$ $(y2) (L' B L') (F2 L B' L') (F2 L2)$ $(y2 z) F2 R U2 R' U2 F2 L' U2 L U2$ $(y z') U2 R U2 R' F2 U2 L' U2 L F2$ $R' U2 R2 U' L' U R' U' L U R' U2 R$ $(y) R2 F2 (R' B' R) F2 (R' B R')$
	$R B' R F2 R' B R F2 R2$ $(x) R D' R U2 R' D R U2 R2$ $(x') R U' R D2 R' U R D2 R2$ $(y' x) R2 D2 R U R' D2 R U' R$ $(y2 x) L U' L D2 L' U L D2 L2$ $(y z) U2 L' U2 L F2 U2 R U2 R' F2$ $(z') F2 L' U2 L U2 F2 R U2 R' U2$ $(y') R' U2 R U' L' U R U' L U R2 U2 R$ $(z) (U R' D r2 U') (R U r2' U') (D')$ $(z' y2) (U' r x) (l2' U L U') (R' U) (L' U' L') (R' U)$
	$x U R' U' L U R U' L' U R U' L U R' U' L'$ $y x' R U' R' D R U R' D2 L' U L D L' U' L$ $x U R' U' L U R U' r2' U' R U L U' R' U$ $y x R U' R' D R U R' u2 R' U R D R' U' R$ $l' U' L' U R U' l' U R' U' L U R U' l' U$ $(r' R' U') (L D' L' U L) (R U' R' D R) U$ $r2 U r2 D x' (R U' R' U)3 x D' r2 U' r2$ $x' U' R U L' U' R' U r2 U R' U' r' F R F'$ $y' z' R' F R2 U R' B' R U' R2 F' R z R B R'$ $R' U L' D2 L U' R L' U R' D2 R U' L$ $(R B L B' R') y (R L) y (L B' R' B L') y' (R' L')$ $l' U' (r' F R F') (R U R' U') L (U R U' R') F$ $(F' R U R' U') L' U R U' (l' U R' U' r) F R$

Permutation of Edges and Corners:

	$(y2) R U' R' U R2 y R U R' U' x U' R' U R U2$ $yz R U R' U' R U2 z' y' R U R' U' y x L' U' L U L2 x'$ $R' U R U' R2 F' U' F U x R U R' U' R2 B'$ $R' U R U' R2' F' U' F U R U' x' R2 U' R' U$ $R' U R U' R2 y' R' U' R U y x R U R' U' R2 x'$ $y R U R' U R U2 R2 U' R U' R' U2 R U r U R' U' L' U R U'$ $R' U2 R' d' R' F' R2 U' R' U R' F R U' F$ $(y) U' R' U R U' R'2 F' U' F U R x U R' U' R2$ $(y2) M U2 r' U l' U2 r U' R2 r U2 R2$
---	---

	(y) R2' u R' U R' U' R u' R2 y' R' U R (y) R2 D B' U B' U' B D' R2 F' U F (y2) F2' D R' U R' U' R D' F2 L' U L R L U2 R' L' y' R' U L' U2 R U' L
	R' U' R y R2 u R' U R U' R u' R2 R' U' R B2 D L' U L U' L D' B2 (y2) L' U' L y' R2' u R' U R U' R u' R2 (y') R' U L' U2 R U' L y R L U2 L' R'
	(y) R2' u' R U' R U R' u R2 y R U' R' (y) R2' u' R U' R U R' u R2 B U' B' (y) R2' D' F U' F U F' D R2 B U' B' (y2) F2' D' L U' L U L' D F2 R U' R' L' R' U2 L R (y) L U' R U2 L' U R' (y') I' U' U' L' U I F' U' L U F R' F R
	(y2) R U R' y' R2 u' R U' R' U R' u R2 (y2) R U R' F2 D' L U' L' U L' D F2 (y2) I2 U' L2 U' F2 L' U' R U2 L' U I x (y) L U' R U2 L' U R' y R' L' U2 R L (y) L U' R U2 L' U R' y' L' R' U2 L R (y') R' F' R F' U' L' U F R' F' L F2 R
	(y2) R' U2 R U R' z' R2' U R' D R U' (y2) L' R' U2 R U R' U2 L U' R U (y2) F2 L' U' r U2 I' U R' U' R2 (y2) R' U2 R U R' U2' L U' R U L' (y2) x U2 r' U' r U2 I' U R' U' R2 B2' R2 U' R2' u R2 D' R2' U R2 R U' L' U R' U2 L U' L' U2' L R L d2 R' U' R d2 R' U L' (L' U R' U2 L U' R) U (L' U R' U2 L U' R) L' U R' U2 L U' R U L' U R' U2 L U' R L' U2 L U L' U2 R U' L U R' L' U' L F L' U' L U L F' L2 U L U
	R U2 R' U' R U2 L' U R' U' L R U2 R' U' R d2' R' U L' U' R y2 R L U2 L' U' L U2 R' U L' U' (R U R' F') (R U R' U' R') F (R2 U' R' U') y' L' U' R U2 L' U L U2 L' R U y2 F2' R U R' y r2 y' R U' R U R2' L' U R U' L U2 R' U R U2' R' y2 r2 U' L' U r' U2 R B' R' U2 (R U' L U2 R' U L) U' (R U' L U2 R' U L) L' U R' U2 L U' L' U2 R L

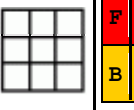
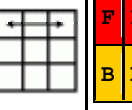
	R' U R2 B2 U R' B2' R U' B2 R2' U' R z D R' U R2 D' R (U' D) R' U R2 D' R U' R y' L U' R U2 L' U R' L U' R U2 L' U R' R U' R' U I U F U' R' F' R U' R U I' U R' L U' L' U L F U F' L' U' L F' L F L' U L' R' U L' U2 R U' L R' U L' U2 R U' L R U' L d2 L' U L R' U' R U2 r' F I' U' y2 L U' R U2 L' U R' L U' R U2 L' U R' U' y z R' U R' D R2' U' R U D' R' D R2' U' R D' R U' L d2 L' U L R' U' R U2 L' U R U' F' R U R' U' R' F R2 F U' R' U' R U F' R'
	R U' R2' F2 U' R F2' R' U F2 R2 U R' z U' R D' R2' U R' (U' D) R D' R2 U R' D R' L' U R' U2' L U' L' R U R' U2' L U' R U' R' U R U' R' F' U' F R U R' F R' F' R U' R z U' R D' R2 U R' U' z' R U R' z R2 U R' z' R U' L' U R' z R2 U R' U z' R U R' z R2 U R' z' R U' L' U L U' r' U' F' U L F L' U L' U' r' U' L L U' R U2 L' U R' L U' R U2 L' U R' U'
	y2 z U R2 U' R2 U F' U' R' U R U F U2 R y' R2 B' R' U' R' U R B R' U2 R U2 R' R U2 R' U2 R B' R' U' R U R B R2' U R U2 R' U2 R B' R' U' R U I U R2 F U2 R' U' R U2 R U2 L' R2 U R U' R L U2 R' y' R U R' F' R U2 R' U2 R' F R U R U2 R' U' R U2 R' U' R' F' R U2 R U2 R' F R U' R' U R U2 R D R' U R D' R' U' R' U R U R' U y2 L U2 L' U2 L F' L' U' L U L F L2 U
	R' U2 R U2 R' F R U R' U' R' F' R2' U' y R2 B2 U' R' U' R U R U B2 R U' R U R' U2 R U' y' R' F R B' R' F' R z x' R' U R' y2 R' U2 I R U' R' U I' U2' R F R U' R' U' R U R' F' R' U2 R' D' R U' R' D R U R U' R' U' R U' (y x) R2 U I U R U' I' U' I U2 R' U2 R
	R U R' U' R' F R2 U' R' U' R U R' F' y2 L' U' L U L F' L2 U L U L' U' L F F R U' R' U R U R2 F' R U R U' R' L2 U' L2 D F2 R2 U R2 D' F2 R2 U' R2 D B2 L2 U L2 D' B2 R2' u' R2 U R2' y R2 u R2' U' R2 R2 U R2' U' R2 U' D R2' U' R2 U R2' D' (z x2) U2 r' U2 r U2 x U2 r U2 r' U2

	$R' U R' U' y x R' U' R^2 x' U' R' U R' F R F$ $R' U R' U' B' D B' D' B^2 R' B' R B R$ $R' U R' U' x^2 y' R' U R' U' I R U' R' U R U$ $y L' U R U' L U L' U R' U' L U^2 R U^2 R'$ $y^2 R U' L' U R' U' R U' L U R' U^2 L' U^2 L$ $y' R' U L U' R U R' U L' U' R U^2 L U^2 L'$ $L U' R' U L' U' L U' R U L' U^2 R' U^2 R$ $R' U R' d' R' F' R^2 U' R' U R' F R F$ $R' U R' U' y z' U' R U' R' U^2 y' R' U' R U R$ $R' U R' U' y R' F' R^2 U' R' U R' F R F$ $R' U R' U' y R' D R' D' R^2 F' R' F R F$ $R' U^2 R U^2 L U' R' U L' U L U' R U L'$ $R' U^2 R U^2 L U' R' U' F' R U' R U' R'$ $R U' L' U R' U' R U' L U R' U^2 L' U^2 L$
	$F R U' R' U' R U R' F' R U R' U' R' F R F'$ $R^2 u R^2' U R^2 D' R' U' R F^2' R' U R$ $F R' F R^2 U' R' U' R U R' F' R U R' U' F'$ $R' U' R F^2 R' U R d R^2 U' R^2' U' R^2$ $R^2 U' R' U R U' z' y' L' U' R U' R' U' L U y z$ $y F' L' U L U L' U' L F L' U' L U L' F' L F$ $F R U' R' F D R' y x R' U' R z R^2 y L' d^2$ $z' U^2 L' U' L U L' y' L' U' R U' R' U' L U$ $F R' F' R U R U' R^2 U' R U I U' R' U$ $z U^2 R U R' U' R y R U L' U L U R' U'$ $F R U' R' U' R d R U R' B' R U' R^2$ $R' F R F' y' U' R' U R^2 U R' U' R' F R F' U'$ $y^2 z U^2 R' U^2 R' U^2 R B R B' U^2 B R' B'$

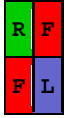
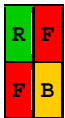
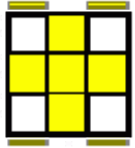
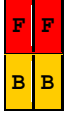
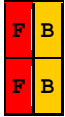
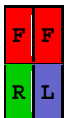
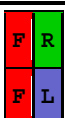
2. COLL+EPPL (ZZ-VH)

This orients and permutes the last layer corners (without affecting edge orientation) in one step, then permutes the last layer edges in the final step.

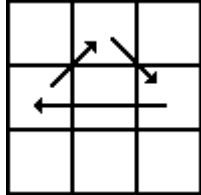
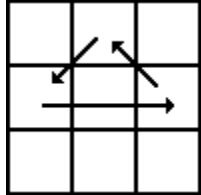
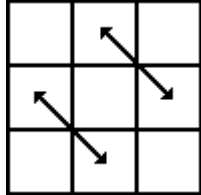
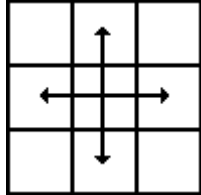
COLL (Lars Vandenberg)

Orientation corners	Permutation corners	Algorithm	Orientation corners	Permutation corners	Algorithm
		$(R U') U' R' U' (R U') (R^2 U') U' R (U R' U) R$			$y' R' U' R F (R^2 D') R (U R' D) (R^2 U') F'$

			(R' U) R2 D r' U2 r D' (R2 U' R)				(R' U) U (R F') (R' F) U2 (F' R) F
			F U' (r' F) R2 F' r U R2 F'				R' U2 R F U' R' U' R U F'
			(R' U) (R U') U' r' (R' F) (R F') r				R' U' (R U') (R' U) U R2 (U R' U) (R U') U' R'
			y' r (U R') U' (r' F) (R F')				(R2 D') (R U') U' (R' D) (R U') U' R
			y' (F R F') r (U R') U' r'				y2 R2 D (R' U) U (R D') (R' U) (U R')
			y (R' U) (U R') D' (R U') U' (R' D) R2				y (R U') (r' F) (R' F) r U (r' F) r
			y2 (R U') U' R D (R' U) U (R D') R2				y2 F (R2 U') R2 U R2 U F' y (R U') U' R'
			R (U R' U) (R U') (R' U) (R U') (R' U) (R U') U' R'				y (R U') U' (R2 U') (R2 U') (R2 U') U' R
			y' F' r (U R') U' (r' F) R				y (R F') U' (R2 F) U' F' U R2 U (F R')
			R' F' (L' F) (R F') r U				(R' U) U R (U R' U) (R2 U') (r' F) R' (F' r)
			(L' U) U (R U') (R' U) U L (R U') R'				y2 (R U') U' R' U' (R U') R2 U (L U') R (U L')
			y R (U R' U) (r' F) (R F') r U2 R'				y' (R U') U' (r' F) R' (F' r) U' (R U') R'
			B' U' R (U R') B U (R U') U' R'				y' L R' U' R U L' U2 (R' U) U R

			$y^2 (F R') U^2 (R F') (R' F) U^2 (F' R)$				$y (R' F) U^2 (F' R) (F R') U^2 (R F')$
			$y' (R U') (r' F) R' (F' r)$				$y (r' F) (R F') r (U R')$
			$y' R (U R' U) (R U') U' R'$				$(R U') U' R' U' (R U') R'$
			$y (R' U) U R U F (R' U) (R U') F'$				$y (R U') U' (R' U) U (r' F) (R F') R' r$
			$R (U R' U) y' (R' U) (R U') (R^2 F) (R F' R)$				
			$(R U') U' R' U' R (U R') U' (R U') R'$				
			$y R (U R' U) R U (r' F) R' (F' r)$				
			$(R B') (R' B) U^2 R^2 F' (r U') r' F^2 R^2$				

EPLL (Lar Vandenberg), permute edges only:

			
$(R' U R') U' R' U' (R' U) R U^2$	$(R^2 U') R' U' R U R U (R U' R)$	$R' U' (R U') R U (R U') (R' U) R U (R^2 U') R'$	$r^2 (R^2 U) r^2 (R^2 U) U r^2 (R^2 U) R^2 r^2$

3. ZBLL (ZZ-a): this method completes the last layer by orienting the corners and permuting the corners and edges, all in one step. It involves 494 distinct cases, and requires learning a minimum of 177 algorithms assuming mirrors and inverses are applied.

F2LL

During the final stages of F2L its possible to manipulate the last layer cubies to reduce the number of LL cases. This partial solving of LL during F2L is called F2LL.

4. ZZ-b or ZZLL

用到 phase 的概念。

Phaseing Strategy: 第二步 F2L 完成最后一个 1X2X2 块 (C/E 块) 时, 还有一个自由度可以利用, 就是在把这对 C/E 块插入下两层时, 调整顶层棱的相对位置, 使颜色相对的楞块处于相对的棱上。完成第二步后, 顶层十字楞块的相对位置或者正确或者只需对调对调其中一对。它可以把 ZBLL 的 494 种顶层状态减少到 167 个, 一步完成顶层只需要 80 个算法, ZBLL 最少需要 177 个。



如图, 顶层一个楞块被 F2L 的 C/E 对 1X1X2 卡住, 但另三个楞块 123 的相对位置可以调整。调整使得左侧卡住楞块的相对色处于相对位置即位置 1。这一步相调整平均需要 5 个动作。

ZZLL 需要的 80 个算法是 ZBLL177 个算法的子集。

ZZLL Algorithms:

<http://stubers.org/jamesstuff/cube/zzll.php>

5. Winter Variation(WV)

This orients the LL corners during insertion of the final 1x1x2 F2L block, reducing the last layer to just PLL.

6. MGLS

On the last F2L slot only the edge is placed. The CLS part of the MGLS method then orients the LL corners, while simultaneously placing the final D-face corner. This leaves only PLL to complete the cube.

7. Blah's Method

This de-oriens the LL corners during insertion of the final 1x1x2 F2L block, allowing completion of LL in one-step using a reduced subset of ZBLL。

Minimal memorization ZZ method

将 ZZ method 的一些中间步骤拆分为多部，以减小记忆量。

E0Line→E0+Line

F2L→3F2L+F2L and Cross Phase

ZZLL→ELL+Orientation of Corners+Permutation of Corners

经过 F2L 阶段的 Phase 调整，有一对楞块相对位置正确，为了利用这一优势而不打乱楞块位置，我们就先把楞块归位，然后再调整角块位置方向，注意不是 COLL。Last layer 和 CFEC 相同，只是状态数要少。

E0-Line 之后转动群就限制为 $\langle L, R, U, F2, B2 \rangle$, 不允许再用 F/B。

最后一步，有一半机会四个棱相对位置完全正确，还有一半机会需要对调两个棱，剩下的就是调整角块了。

最后一步保持楞块位置朝向不动，调整角块，没有现成公式。

第十二节 ZB Method

ZB Method 是 Zbigniew Zborowski 和 Ron van Bruchem 2002 年各自独立发展出来。是 Fridrich 方法的一个高级变种。此方法的特点是在还原最后一对 F2L 时，调对顶层楞块方向，到最后一层时，大大削减了顶层状态数，最小有 176 个公式，因此可以一步还原顶层。可以简单对比一下 Bernard Helmstetter 方法，Helmstetter 前两层正常的做 F2L，第三层将 OLL，P11 合并为一步，有 1211 个公式。

即便如此，ZB method 的公式量依然很大。ZBF2L 这一步有 306 中状态，ZBLL 这一步有 494 种状态。这两步各自最小需要 124 和 176 个公式，总共是 300 个公式。这是它的缺点，同样也是它优点，利用大量公式减少了转动步数，提高了还原速度。

The table below shows the number of formulas required for solving separate steps of ZB system:

	number of formulas	number of formulas with reverse ones	number of formulas with reverse ones and symmetrical ones
First step	none	none	none
Second step	125 (minus 1 solved)	158 (minus 1 solved)	306 (minus 1 solved)
Third step	177 (minus 1 solved)	270 (minus 1 solved)	493 (minus 1 solved)
Total	300		

一. F2L without one edge and one corner

解决 F2L, 留下一个 Corner/Edge 对。

二. F2L last edge and corner and last four edges orientation

解决 F2L 最后一对 C/E 的同时，调整好顶层 4 个楞块的朝向。这一步有 125 个公式。

三. last four edges permutation and last four corners orientation/permutation

顶层棱块位置调换，角块方向扭转和位置调换，都在一步完成。有 177 个公式。

ZBLL Algorithms:

(待续)

第十三节 VH Method

由 Lars Vandenberg 和 Dan Harris 发展，Fridrich 方法的变种。

一. Cross

二. F2L (3pairs)

三. 4th F2L pair+Orientation of LL Edge

In the VH System, when solving the 4th F2L pair you also Orient the edge pieces of the Last Layer. You do this in two stages, called "pair-up", and "insert". The pair-up and insert method for F2L is nothing new, but in the VH system you use a special set of algorithms to flip the necessary edges during the "insert" stage. There are 32 special algorithms, and by using these you can then go straight into the next stage.

注意这一步与 ZBF2L 的不同。F2L 可以分解位配对 C/E 块和将其插入一二层这两个连续动作。VHF2L 就是在最后一对 F2L 的插入下两层时为顶层楞块对色。配对之后大大减少了状态数，所以只有 32 个公式。而 ZBF2L 是在 C/E 块配对之前为顶层棱对色，最小需要 124 个公式。因为要先配对，VHF2L 这一步公式的减少是以增加多余步数为代价的，所以不适合竞速。

VHF2L's Algorithms:

Algorithms for Connected Pairs - solving into the FR slot (Red is the F colour, Green is the R colour)			
Pair is locked against a correctly oriented edge			
Case PC-UR_UFR-000	Case PC-UR_UFR-001	Case PC-UR_UFR-010	Case PC-UR_UFR-011
$U R U' R'$	$U (F' U F)(R U' R')$	$U (F' U^2 F)(R U' R')$	$U^2 (R U' R')(F' U F)(R U' R')$
Pair is separte from correctly oriented edges			
Case PC-UR_UFR-100	Case PC-UR_UFR-101	Case PC-UR_UFR-110	Case PC-UR_UFR-111

$R' F R F'$	$U2 (R U' R') U' (R' F R F')$	$U (R U R') U2 (R' F R F')$	$d (R' U2 R) d (L U' L' U') (L' B L B')$
Pair is locked against a correctly oriented edge			
Case PC-UF_UFR-000	Case PC-UF_UFR-001	Case PC-UF_UFR-010	Case PC-UF_UFR-011
$d' (L' U L)$	$U' (R U' R') (F' U F)$	$U' (R U2' R') (F' U F)$	$d' U' (L' U L) (F' U' F') (L' U L)$
Pair is separate from correctly oriented edges			
Case PC-UF_UFR-100	Case PC-UF_UFR-101	Case PC-UF_UFR-110	Case PC-UF_UFR-111
$F R' F' R$	$d U (R' U R U) (R B' R B)$	$d' (L' U' L U2') (L F' L' F)$	$U (R U R') (F' U F) d (R B' R' B)$

Algorithms for Separated Pairs - solving into the FR slot (Red is the F colour, Green is the R colour)			
Case PS-UB_UFR-000	Case PS-UB_UFR-001	Case PS-UB_UFR-010	Case PS-UB_UFR-011
$R U' R'$	$(R U' R' U') (R U R' U2) (R' F R F')$	$(R U' R') U2 (R' F R F')$	$(R U2' R') (F' U2 F) (R U' R')$
Case PS-UB_UFR-100	Case PS-UB_UFR-101	Case PS-UB_UFR-110	Case PS-UB_UFR-111

$(R\ U2'\ R'\ U')\ (R'\ F\ R\ F')$	$(R\ U2'\ R')\ (F'\ U\ F)\ (R\ U'\ R')$	$(R\ U'\ R'\ U')\ (F'\ U\ F)\ (R\ U'\ R')$	$(R\ U'\ R')\ (F'\ U\ F\ U2)\ (R\ U\ R'\ U2)\ (R'\ F\ R\ F')$
Case PS-UL_UFR-000	Case PS-UL_UFR-001	Case PS-UL_UFR-010	Case PS-UL_UFR-011
$F'\ U'\ F$	$(F'\ U'\ F2\ U)\ (R\ U'\ R'\ F')$	$(F'\ U\ F\ U2)\ (F\ R'\ F'\ R)$	$U2\ (F'\ U')\ (L'\ U\ L\ U\ F)$
Case PS-UL_UFR-100	Case PS-UL_UFR-101	Case PS-UL_UFR-110	Case PS-UL_UFR-111
$(F'\ U2\ F\ U)\ (F\ R'\ F'\ R)$	$(F'\ U2)\ (L'\ U\ L\ F)$	$(F'\ U2\ F)\ (R\ U2'\ R')\ (F'\ U\ F)$	$F\ D\ B'\ R'\ B\ D'\ F2\ U'\ F$

四. COLL

provided you have managed to successfully Orient the LL edges correctly, you will arrive at one of the 40 COLL cases. During COLL you both Orient and Permute the LL corners simultaneously, whilst keeping the LL edge orientation intact. This results in a complete top face, and in 1/12 instances on average, a completed Last Layer! These are dramatically reduced odds from the chances of finishing the Last Layer in 1 step in the standard CFOP method. In the other 11/12 times, you only have a simple and fast edge permutation to solve, since the corners are already permuted.

一步还原角块位置和方向。

Corners and Orientation Last Layer

This is a list of all 40 algorithms I use to solve the corners of the last layer without affecting the orientation of the edges. I tried to note down the moves in such a way so that you can see how I perform them. Moves that are between brackets are move sequences that are performed in one single movement.

Orientation corners	Permutation corners	Algorithm	Orientation corners	Permutation corners	Algorithm
		$(R' U') U' R' U' (R U') (R2 U') U' R (U R' U) R$			$y' R' U' R F (R2 D') R (U R' D) (R2 U') F'$
		$(R' U) R2 D r' U2 r D' (R2 U' R)$			$(R' U) U (R F') (R' F) U2 (F' R) F$
		$F U' (r' F) R2 F' r U R2 F'$			$R' U2 R F U' R' U' R U F'$
		$(R' U) (R U') U' r' (R' F) (R F') r$			$R' U' (R U') (R' U) U R2 (U R' U) (R U') U' R'$
		$y' r (U R') U' (r' F) (R F')$			$(R2 D') (R U') U' (R' D) (R U') U' R$
		$y' (F R F') r (U R') U' r'$			$y2 R2 D (R' U) U (R D') (R' U) (U R')$
		$y (R' U) (U R') D' (R U') U' (R' D) R2$			$y (R U') (r' F) (R' F) r U (r' F) r$
		$y2 (R U') U' R D (R' U) U (R D') R2$			$y2 F (R2 U') R2 U R2 U F' y (R U') U' R'$
		$R (U R' U) (R U') (R' U) (R U') (R' U) (R U') U' R'$			$y (R U') U' (R2 U') (R2 U') (R2 U') U' R$
		$y' F' r (U R') U' (r' F) R$			$y (R F') U' (R2 F) U' F' U R2 U (F R')$
		$R' F' (L' F) (R F') r U$			$(R' U) U R (U R' U) (R2 U') (r' F) R' (F' r)$
		$(L' U) U (R U') (R' U) U L (R U') R'$			$y2 (R U') U' R' U' (R U') R2 U (L U') R (U L')$

			$y R (U R' U) (r' F) (R F') r U^2 R'$				$y' (R U') U' (r' F) R' (F' r) U' (R U') R'$
			$B' U' R (U R') B U (R U') U' R'$				$y' L R' U' R U L' U^2 (R' U) U R$
			$y^2 (F R') U^2 (R F') (R' F) U^2 (F' R)$				$y (R' F) U^2 (F' R) (F R') U^2 (R F')$
			$y' (R U') (r' F) R' (F' r)$				$y (r' F) (R F') r (U R')$
			$y' R (U R' U) (R U') U' R'$				$(R U') U' R' U' (R U') R'$
			$y (R' U) U R U F (R' U) (R U') F'$				$y (R U') U' (R' U) U (r' F) (R F') R' r$
			$R (U R' U) y' (R' U) (R U') (R^2 F) (R F' R)$				
			$(R U') U' R' U' R (U R') U' (R U') R'$				
			$y R (U R' U) R U (r' F) R' (F' r)$				
			$(R B') (R' B) U^2 R^2 F' (r U') r' F^2 R^2$				

五.EPLL

调对楞块位置。只有四种情况：

Permute edges

$(R' U R') U' R' U' (R' U) R$ $U R^2$	$(R^2 U') R' U' R U R U (R$ $U' R)$	$R' U' (R U') R U (R U') (R'$ $U) R U (R^2 U') R'$	$r^2 (R^2 U) r^2 (R^2 U) U r^2$ $(R^2 U) R^2 r^2$

这个是 Vandenberg 的算法。

EPLL 是 PLL 的一个子集。还有其他算法。

第十四节 Makisumi-Garron Last Slot (MGLS)

Shotaro Makisumi, Lucas Garron, 2007 年发展。

Step 1. F2L, minus last slot

底下两层，留一对 C/E

Step2. ELS, Edge step of the Last Slot

The edge of the last slot is placed while the last layer edges are oriented.

植入中间层楞块，同时调整底层楞块方向。

Step3. CLS, Corner step of the Last Slot

The corner of the last slot is placed while the last layer corners are oriented

104 cases/56 algs+mirrors, ≈ 9.5 twists (average)

植入底层角块同时调整所有顶层角块方向。

Step4. PLL, Permutation of the Last Layer

21 cases/14 algs+mirrors, ≈ 12 twists (average)

第十五节 Thistlethwaite method

Section 1. Heise's Human Thistlethwaite Method

$G_0 = \langle U, D, L, R, F, B \rangle$, Phase 1 $\rightarrow$
 $G_1 = \langle U, D, L, R, F_2, B_2 \rangle$, Phase 2 $\rightarrow$
 $G_2 = \langle U, D, L_2, R_2, F_2, B_2 \rangle$, Phase 3 $\rightarrow$
 $G_3 = \langle U_2, D_2, L_2, R_2, F_2, B_2 \rangle$, Phase 4 $\rightarrow$
 $G_4 = \langle I \rangle$

普通解法是通过逐块（piece or block or layer）还原来减少下一步剩余块的状态数，最后所有块还原。Thistlethwaite 方法（TM）则与此有本质的不同。魔方的任何一种状态与魔方群的一个群元素是一一对应的。TM 的思想就是逐步降解魔方所处的群到更小的子群，最后到单位子群，也即还原状态。所以在还原的每一步实体魔方看起来还是乱的，但实际上魔方的状态数是随所处的群的减小而规则的减小的。

这个表是魔方在相应子群时的状态数：

Group	positions	Factor
$G_0 = \langle U, D, L, R, F, B \rangle$	$4.33 \cdot 10^{19}$	
Phase 1		$2,048(2^{11})$
$G_1 = \langle U, D, L, R, F_2, B_2 \rangle$	$2.11 \cdot 10^{16}$	
Phase 2		$1,082,565(12!/(8!4!) \cdot 3^7)$
$G_2 = \langle U, D, L_2, R_2, F_2, B_2 \rangle$	$1.95 \cdot 10^{10}$	
Phase 3		$29,400(8!/(4!4!))^2 \cdot 2 \cdot 3$
$G_3 = \langle U_2, D_2, L_2, R_2, F_2, B_2 \rangle$	$6.63 \cdot 10^5$	
Phase 4		$663,552(4!^{5/12})$
$G_4 = \langle I \rangle$	1	

Phase 1 对好所有楞块方向

从 G_0 到 G_1 有个缩小因子 $2,048(2^{11})$ ，它表示在这一步魔方状态数减少了 2048 倍。因为楞块有 2^{11} 种朝向，这一步就是调整楞块到某一种朝向模式。原因是，若要翻转一个楞块，必须利用到 FRU 三个转动的组合，但是经过 Phase1 调整后，不能有 F 操作（只能有 F^2 ），也即所有楞块方向不能在翻转。所以才有下面这个定义：

定义：如果能通过转动群 $\langle U, D, L, R, F^2, B^2 \rangle$ 使得某个楞块还原，那么这个楞块朝向就是好的。反之则错。

Phase1 就是将所有朝向错的楞块变好。方法很简单，首先找到所有朝向错误的楞块，然后把四个朝向错的楞块通过 $\langle U, D, L, R, F^2, B^2 \rangle$ 转动移到 F 面（或者 B 面）上来，做一个 F 转动，这四个则变为好的。如果只有两个错的，把其中一个移到 F 面，做一个 F 转动，这个错的变为对的，F 面上另三个对的变为错的。这样就凑足了四个错的。

熟悉 ZZ 玩法 EOLine 的朋友对这一步应该很熟。

下面为了叙述方便，我把魔方放置为：顶层黄色，底层白色，左面蓝色，右面绿色，前面红色后面橙色。我把黄色白色合称 YW 色，蓝色绿色合称 BG 色，红色橙色合称 RO 色。说道某个面是 YW 色时就指这个面上可以有黄色白色两种颜色。说到某个块的贴片是 YW 色时就是指这个贴片可以是黄色也可以是白色。其他类推。

Phase 2 将中间层楞块放到中间层，并调整好角块朝向

注意这一步魔方已经处于 G_1 子群了，在这一步的所有操作只能由 G_1 群 $\langle U, D, L, R, F^2, B^2 \rangle$ 生成，把魔方降解到 G_2 群 $\langle U, D, L^2, R^2, F^2, B^2 \rangle$ 。这一步群的大小减小也即魔方的状态数减小 $3^7 * 12! / (8!4!)$ 倍。 3^7 对应于角块方向， $12! / (8!4!)$ 表示中间层楞块放到中间层。它的物理原因是，任何两个相邻的角块通过 $\langle U, D, L^2, R^2, F^2, B^2 \rangle$ 随意转动，当把它们再次转为相邻时，它们的三个贴片相对方向不变。也就是 G_2 群固定了角块到某一个特定朝向模式。同样， $\langle U, D, L^2, R^2, F^2, B^2 \rangle$ 生成的转动不可能把上下层转到中间层。这就是要求这一步要把中间层的楞块放回中间层。再调整角块朝向之后，U 面或者 D 面只有黄白两种颜色。

下面说具体怎么做。

中层楞块放回中层：等价的说法就是要把 YW 色楞块全部放在顶底面。通过 L, R 转动，使得顶面有三个楞块是 YW 色，底面也有三个楞块的 YW 色朝下。他们缺的那块转到 L/R 侧面，剩下两个 YW 色楞块必定在中间层，也转到这个侧面，做一个 L/R 转动，中间层楞块全部回到中间层。

此后，除非必要，便不可在用 LR 操作，免得移走中间层楞块。如果要用的 L, R 转动的话，L/R 之后即需跟上自己的逆。用数学表示就是 $R\{U, D, L^2, R^2, F^2, B^2\}R'$ 这样的转动。

调整角块朝向：就是使角块的 YW 色处于 U 或者 D 面上。把两个朝向错误的角块一个放在 LUF，一个放在 FDR， $R'D-L^2-D'R$ 操作即可以把它们的 YW 色调到 U/D 面上。也可以放在 RUF, LFD 两个位置，用操作 $LD'-R^2-DL'$ 翻转它们的朝向。

Phase 3 调整 F/B 面为 RO 色，L/R 面为 BG 色，并调整角块相对位置

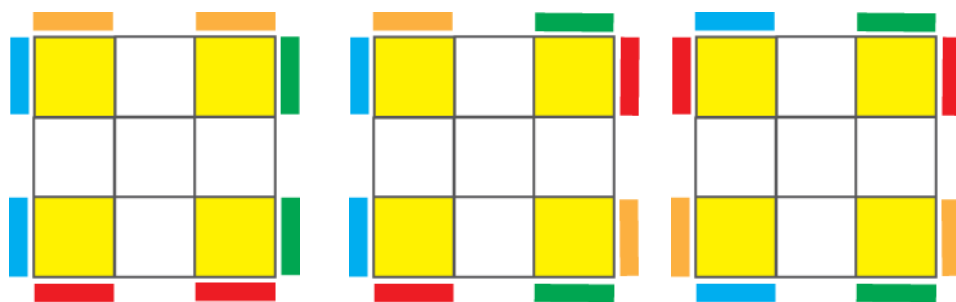
这一步所有操作只能通过 G2 群的转动<U,D,L2,R2,F2,B2>生成。

上一步把 U/D 调整为 YW 色，这一步类似，要调整两个侧面为对色。或者换一个说法，上一步已经把 E 层（中间层）楞块放回中间层了，这一步就是把 M 层（M slice）上的楞块调到 M 层，S 层（S slice）上的楞块调到 S 层。显然这是 G3 群<U2,D2,L2,R2,F2,B2>决定的状态。但这还不够，G3 群还决定了角块的相对位置。八个角块分为两组，第一组四个角块{ULB, URF, RDB, LFD}在{ULB, URF, RDB, LFD}这四个顶角位置置换，另一组四个角块{ULF, URB, LDB, RDF}在{ULF, URB, LDB, RDF}这四个顶角位置置换。比如，如果 URB 角块处于 URF 顶角上，它的位置就是错的，要处于 RDF 顶角上则是对的。这一步使状态数减少了 29,400 ($= [8!/(4!4!)]^2 * 2 * 3$) 倍， $8!/(4!4!)$ 对应于 M 层 S 层楞块放回 M 层 S 层，6 对应于两组角块每一组的四个相对位置正确。



下面介绍怎么做到这一点。

角块调整难度大一些，要首先调整角块位置。可以直接在上面两组对角线上找，看是不是该组角块处于该组，不是则调为是。但这样比较麻烦。简单的做法是，先黄色角块全部调到 U 层，同时白色角块都回到 D 层。这时，看 U/D 层各自的四个角块相对位置是不是正确。比如 U 层角块，有三种状态，



第一种侧面颜色完全匹配；第二种不匹配，有且只有一个侧面同色；第三种完全不匹配。底面同样判断。

如果 U 和 D 面都是完全匹配或者都是完全不匹配（一三），（操作 R2F2R2 可以使情况一三互相变换），那么两组角块的相对位置都正确。否则需要调整。

R'FR' - B2 - RF'R，这个操作交换 U 层 ULB, URB 两个角块，同时交换 D 层 DLB, DRF 两个角

块。转动<U,D,L2,R2,F2,B2>使角块处于合适位置，用这个互换操作即可把两组角块相对位置调整正确。最多两次即可调好。比如，经常出现的情况是上下两层各有一对匹配的，将其中一对匹配的比如 U 层的对调，结果就是 U 层四个角块就变为完全不匹配，D 层那对匹配的打乱出现新的一对匹配的。这时，对换 U 层一对对角块，对换 D 层那对不匹配的（相邻的），则上下两层都变为完全匹配。

接着处理楞块。注意这时要把八个角块放到（相对）正确位置（即处于 G3 群转动允许的的位置），之后便不能用 U，D 转动，或者只能这么用 $U\{U2,D2,L2,R2,F2,B2\}U'$ 。处理楞块，首先把四个 RO 色楞块调到 D 层，把四个 BG 色楞块调到 U 层。D'-M2-D 是将当前 S-slice 在 D 层的两个楞块放到末态 M-slice 的 U 层，将当前 M-slice 在 U 层的两个楞块放到末态 S-slice 的 D 层。

有一种情况是 M-slice 上面只有一个 S-slice 的楞块，这时，用 D'-M2-D，让 M-slice 上有两个 S-slice 的楞块。之后再用上面的方法。

群被降解到了 G3。

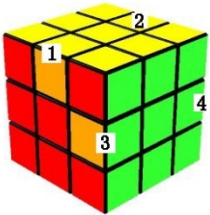
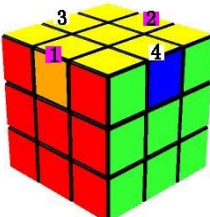
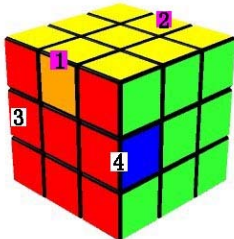
Phase 4 还原所有角块楞块

这时只能用 G3 群的转动<U2,D2,L2,R2,F2,B2>。同理，如果必要用到某个 90 度角转动如 R，必须跟着它的逆， $R\{U2,D2,L2,R2,F2,B2\}R'$ 。这一步所需的技巧很简单，如果想练得话，可以把还原好的魔方用转动{U2,D2,L2,R2,F2,B2}打乱，再还原，掌握这一步所需的技巧。熟练之后全凭直觉即可还原。

最直观的做法先还原上下两层，首先还原两个相邻角块，这时剩下六个角块都自动对好还原。用公式 $U2M2U2$ 对调 M 层上两对楞块，用公式 $U2R2U2R2U2R2$ 将 M 层一对楞块和 E 层一对楞块对调。其他情形都可以变为这两种。

下面具体点用图说话：

Case1		MD2M'
Case2		M2U2M2 UF<-->UB, DF<-->DB

Case3		U2R2U2R2U2R2 1<-->2, 3<-->4
Case4		M2U2M2-U (M2U2M2) U' 1<-->2, 3<-->4
Case5		E-U2R2U2R2U2R2-E' 1<-->2, 3<-->4

Case 1. 还原一个棱

MD2M'

Case2. 对调 M 层上的两个棱，UF<-->UB, DF<-->DB

M2U2M2

Case3. 对调 M 层上两个棱，1，2；同时对调 E 层上两个棱 3，4。

U2R2U2R2U2R2

剩余所有情形都可以转化为 Case2，Case3。比如

Case4. 对调 M 层 1，2 楞块，对调 S 层 3，4 楞块。可以先 M2U2M2，这时就转化为 case2，

接着转动 U (M2U2M2) U'。 M2U2M2-U (M2U2M2) U'

或者直接 M2 把 1，2 转到底层，再用 case2 也可。

Case5. 对调 M 层 1，2 块，E 层 3，4 块。先做个 U 转动，就转化为 case3.按照 case3 处理，

最后 U'。或者转动 E 变为 Case3，最后 E'。

要对掉的棱处于 M，S，E 的对角位置，处理方法类似，都是转化为 2，3 两个基本转动。

用上面公式，假如已经还原好两个面，比如 case2，还原好 U，D 面。剩下 E 层上的棱需要调整，这时把魔方整体转动，UD 面转做左右面，E 层转为 M 层。之后用 U2{M2，M}U2 即可还原整个魔方。

补充说明一下，上述过程中比如第四步的 Case4，用到了 U，U'转动，是为了为了简便打破了暂时打破 G3 之后立马复原。实际上这些操作完全可以只用 G3 群生成元{ U2,D2,L2,R2,F2,B2} 实现。Case4:

M2U2M2U2--y'--U2R2U2R2U2R2---x2y--U2R2U2R2U2R2—变成 E 层三个棱的轮换
F2U2R2U2 F2U2R2U2 轮换这三个棱。

再比如第三步调整角块位置复原，我们用了 $R'FR' - B2 - RF'R$ ，这个操作交换 U 层 ULB，URB 两个角块，同时交换 D 层 DLB，DRF 两个角块。这一步也可以在 G2 群内实现。这么操作：
UF2U' R2UF2U' F2-L2D' L2

它的效果就是对换 (ULF, UBL) + 轮换 (UF, UL, DB, DR)，后面这个棱块轮换可以用 D' M2D 等简单操作调整到位，先不用管它。反复对换角块即可把所有角块归位。

降群法加速

第一步：调整棱块朝向

可以顺手在底面做个十字，四步左右。剩下 8 个棱块，方便判断。

棱块朝向的快捷判断方法：

一个楞块由两个颜色标记，它所处的位置可由两个面或两个色标记，所以两对有序数组可以完全描述一个楞块的状态。面（色）在{ U2,D2,L2,R2,F2,B2}转动下是等价的，只需要对三个数来编码颜色，上下面色=0，前后色=1，左右色=2。下面我把魔方这么放置

上黄下白，前红后橙，左蓝右绿。白黄=0，红橙=1，绿蓝=2。那么红白块的颜色编码就是 10，它在蓝黄面上，则位置编码为 20。能被转动{U,D,L,R,F2,B2}还原的是对的，记为 0，反之记为 1。

下表是六种色块在六种位置上的对错表。

色 位	01	02	12	10	20	21
01	0	0	0	1	1	1
02	0	0	0	1	1	1
12	0	0	0	1	1	1
10	1	1	1	0	0	0
20	1	1	1	0	0	0
21	1	1	1	0	0	0

有两组数，{01，12，02}，{10，20，21}，当楞块的颜色和位置在同一组时就可以还原，分别在不同组则不能被还原。或者说楞块两个贴片颜色的大小顺序和这两个贴片位置的大小顺序相同时它的朝向是对的，相反则是错的。

定义：上下面为高级面，前后面为中级面，左右面为低级面。

我的魔方这么放置，上黄下白，左蓝右绿，前红后白，那么，高级面相应的黄白色就定义为高级色，红橙定义为中级色，蓝绿定义为低级色。

判断：如果一个棱块的级别高的颜色处在级别高的面上，它的朝向就是正确的。如果它的级别低的颜色处在级别高的面上，它的朝向就是错的。

第二步：上下面颜色复原

上下面 Cross 完成之后，调整角块朝向。一次调正上面角块朝向，再一次调正底面角块朝向。

有七个公式：

OLL 21 	$(RUR')U(RU'R')U(RU2R')$	OLL 22 	$RU2-R2U'-R2U'-R2U2 R$ $RU'2(R2U')(R2U')(R2U'2)R$
OLL 23 	$R2-D'-RU2R'-D-RU2R$	OLL 24 	$r-U R'U'-r'-FR F'$ $r(U R')U'(r' F)(R F')$
OLL 25 	$F'rU- R'-U'r'F- R$ $F' r(U R')U'(r' F) R$		
OLL 27 	$(RUR')U (RU2R')$	OLL 26 	$(RU2R') U'(RU'R')$

第三步：角块位置，楞块位置

角块位置

把匹配的那一对角块转动到 B 面，一次调整好八个角块相对位置。

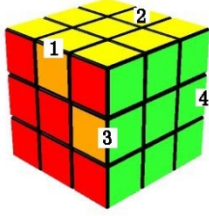
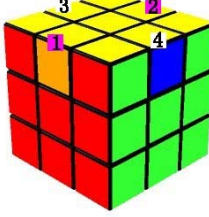
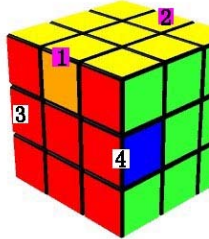
U0 D0	R2F2R2
U1 D0	(R' U R') B2 (R U' R)
U1 D1	(R2 U)(F2 U2 R2)(U R2)
U1 D4	(R U' R) F2 (R' U R) (F2 R2)
U0 D4	(R U' R2)(F2 U' R) F2(R' U F2) (R2 U R')

Note: U1 D1 表示上下面各有一对匹配的角。

楞块位置：D'M2D 对调 S-slice 上的两个楞块到 M-slice。假如 M-slice 上只有一个 S-slice 上的楞块，可以用 MD2M'，即第四步 Case1 的公式把这个棱放回 S-slice。

第四步：180 度转动还原魔方

Case1 	MD2M'
-----------	-------

Case2		M2U2M2 UF<-->UB, DF<-->DB
Case3		U2R2U2R2U2R2 1<-->2, 3<-->4
Case4		M2U2M2 -U (M2U2M2) U' 1<-->2, 3<-->4
Case5		E-U2R2U2R2U2R2-E' 1<-->2, 3<-->4

Section 2 Jaap's Description

Jaap's Puzzle Page

Thistlethwaite's method differs from layer algorithms and corners first algorithms in that it does not place pieces in their correct positions one by one. Instead it works on all the pieces at the same time, restricting them to fewer and fewer possibilities until there is only one possible position left for each piece and the cube is solved.

The way it does this is by first doing a few moves until a position arises that can be solved without using quarter turns of the U and D faces (though half turns of U and D are still needed). It then proceeds to solve the cube without using U or D quarter turns by first moving to a position that does not need quarter turns of the F, B faces either. With these further restrictions a position is arrived at that does not need any quarter turns at all, and can hence be solved by half turns only. The cube then indeed gets solved using half turns only.

These four stages are quite complicated, since they use large look-up tables for all the positions at each stage. The table below shows the numbers involved:

Group	# positions	Factor
$G_0 = \langle L, R, F, B, U, D \rangle$	$4.33 \cdot 10^{19}$	$2,048(2^{11})$
$G_1 = \langle L, R, F, B, U^2, D^2 \rangle$	$2.11 \cdot 10^{16}$	$1,082,565(12C4 \cdot 3^7)$
$G_2 = \langle L, R, F^2, B^2, U^2, D^2 \rangle$	$1.95 \cdot 10^{10}$	$29,400(8C4^2 \cdot 2^3)$
$G_3 = \langle L^2, R^2, F^2, B^2, U^2, D^2 \rangle$	$6.63 \cdot 10^5$	$663,552(4!^5/12)$
$G_4 = I$	1	

Mathematically speaking, this is a sequence of nested groups, and each stage of the algorithm is simply a look-up table showing a solution for each element in the quotient coset space. The last column shows the order of these coset spaces, i.e. the size of the look-up tables used.

The first stage has a factor of $2,048=2^{11}$. This corresponds to the fact that this stage fixes the orientation of the edges; it is impossible to flip edges if only half turns are allowed on the U, D faces, but otherwise the pieces can be put in any normal position with this restriction.

注意，翻转一个楞块必须 FUR 三个动作都用到才能实现。从 G_0 到 G_1 子群的物理含义就是不再翻转楞块。

The second stage has a factor of $3^7 * 12! / (8!4!)$. This corresponds to the fact that this stage fixes the orientation of the corners, and places the middle layer edges into their slice. It is fairly easy to see that if you are only allowed half turns on the F,B,U,D faces together with any turns on the L, R faces, then the left and right faces will never have more than two colours, and the middle edge pieces will never leave their slice or be flipped. In fact, those restricted moves can solve any position under those conditions.

这一步是固定角块朝向 (3^7)。即角块 L, R 贴片只能在 L, R 面上。同时把中间层上的楞块放到中间层 ($12! / (8!4!)$)。G2 群不可能把 LR 层上的楞块带到中间层。

The third stage has a factor of $[8! / (4!4!)]^2 * 2 * 3$. This corresponds to the fact that the edges in the L and R faces are placed in their correct slices, the corners are put into their correct tetrads, the parity of the edge permutation (and hence the corners too) is made even, and the total twist of each tetrad is fixed.

这一步分别把 LR 面上的角棱放到 LR 面，楞块（角块）调整成偶置换状态，LR 面上的角块的 twist 确定下来。

The final stage finally solves the cube.

The following table shows the number of moves each stage needs.

	1	2	3	4	Total
Original algorithm:	7	13	15	17	52
Improved algorithm:	7	13	15	15	50
Best possible:	7	10	13	15	45

Through extensive calculations, Mike Reid has shown that going from G_0 directly to G_2 can be done in 12 moves, and from G_2 to G_4 in 18 moves. When combining these, the 30-move cases can be avoided, so it has been proved that 29 moves is always sufficient to solve the cube. Silviu Radu has recently shown that the cases apparently needing 29 moves can be solved in fewer, so that in fact 28 moves suffice.

[Ryan Heise](#) has developed a way of solving the cube based on Thistlethwaite's algorithm. He splits stages 2, 3 and 4 into two steps each (corners and edges separately) in order to make it possible for a human being to memorise.

Section 3. Heise: Human Thistlethwaite Algorithms

Phase 1 -> <U,D,L,R,F2,B2> group

- simple, no algorithms

Phase 2 -> <U,D,L2,R2,F2,B2> group

- Direct up/down edges to up/down face (simple, no algs)
- Direct corners to up/down face (between 8 and 60 algs)

Phase 3 -> <U2,D2,L2,R2,F2,B2> group

- Corners (between 1 and 2 algs)
- Edges (between 1 and 4 algs)

Phase 4 -> place pieces

- Corners (intuitive)
- Edges (intuitive)

Step 1 (Oriente the Edges)

第一步调整了所有楞块朝向。

Put cube into <L2,R2,F,B,U,D> group, which I will call G1. That is to say, put the cube into a position where the cube can be solved by only using L2,R2,F,B,U,D moves. It helps in this method to think of opposite colours as the same colour up until you get to step 4. Let's call the left/right colours "green" (these are unimportant in step 1), the up/down colours red, and the front/back colours white.

Once you get into G1, it is impossible to flip edges, and the only thing we need to do to get the cube into G1 is to make sure all the edges are flipped right. First you need to find bad edges that are flipped the wrong way. A bad edge is one that, if moved to its home position using only moves from G1, will be flipped the wrong way. (Of course, you don't need to move it there to see that)

There are always an even number of bad edges, between 0 and 12, or 6 on average. The only way to flip edges to turn the left or right sides 90 degrees, and that will flip the 4 edges on that side. To flip 4 bad edges, you simply move them all into positions on the left side (for example), and turn that side 90 degrees. If you have only 2 bad edges, you can make that into 4 by moving just one of them onto the left side and turning it. That will change one of the bad edges into 3 (plus the other one it makes 4).

At end of step 1, you should see no red edges facing to the front or back, and you should see no white edges facing to the top or bottom. To the untrained eye, it will look like you have accomplished nothing so far! Yet, these 12 edges are in a much better position now.

step 2

第二步使得 UD 面上的所有棱块都在 UD 面上，自然地，第二层上的棱都在第二层。同时调整角块朝向。使角块的 U/D 贴片在顶面或底面即可。

Put cube into $\langle L2, R2, F2, B2, U, D \rangle$ group, which I will call G2. In this group, pieces in the middle layer will stay in the middle layer without flipping, and pieces in the top and bottom layers can only exchange between those layers without twisting or flipping. At the end of step 2, you should only see red on the top and bottom. In the middle layer, you should only see green on the left/right sides, and white on the front/back sides.

Remember we are currently in G1, so to get to G2, we can only use moves from G1. To start with the edges, we want to get a red cross on the top and bottom. There are 8 red edges and because of limited space, at least 4 of those must already be forming part of the crosses. The general strategy is to first get 3 red edges on the top and 3 edges on the bottom, and align them so that the missing parts of both crosses are facing to the front. The other 2 red edges must be somewhere in the middle layer (if not, you must have 4 edges instead of 3 edges on the top or bottom). Using double turns around the $L2, R2, F2, B2$ sides, move those two remaining red edges over to the two front positions on the middle layer, and turn the front 90 degrees. With the edges done, we have created some symmetry so it is possible rotate the cube about the U-D axis and change your perception of what is the front and back. Very convenient.

Now for the corners, we can twist two corners at a time. Start with one corner at LUF (corner A) and the other corner at FDR (corner B). Then do:

R'D - twists the B corner
L2 - swaps the B and A corners
D'R - twists the A corner on the way back

Do this repeatedly until all corners are twisted the right way. You should have all red on the top and bottom.

step 3

第三步，使所有相对色在相对面上。即调整使 F/B, L/R 面上只有相对色。但这还不够。

Put cube into $\langle L2, R2, F2, B2, U2, D2 \rangle$ group, named G3. In the previous step, you put only opposite colours on the top and bottom (red). In this step, you do the same for the front/back

(white) and left/right (green). In G3, each corner can only move between one of 4 positions, which I call a circuit. There are only two circuits and each corner belongs to one of those circuits. Within G3, the only moves you can do (double turns) will swap two corners in one circuit, and two corners in the other circuit that overlap. This limits the permutations that are possible, and it means that simply getting opposite colours on all sides is not enough to satisfy G3.

Since the corners are the tricky bit here, we will put the corners into G3 first, then the edges. You must now distinguish between opposite colours to know if G3 is satisfied. I will use red opposite to orange, white to yellow, and green to blue. An easy way to see if G3 is satisfied is to move all red corners to the top side, and all orange corners to the bottom side. Pairs of corners around the top and bottom will either match or mismatch. If they either all match, or they all mismatch, the corners are in a G3 position. There is only one way for all corners to match, and one way for all corners to mismatch (you swap a pair of opposite corners), and so when the exact pattern occurs on the top and bottom, the bottom corners should align exactly with the top corners. You can switch between all matched, and none matched, by doing R2U2R2 (that swaps a pair of opposite corners on the bottom, and a pair of opposite corners on the top). If you're not in either of those positions, here is a sequence that will swap a pair of opposite corners on the bottom, and an adjacent pair of corners on the top:

R'FR' - move the 4 target corners to the back side

B2 - swap!

RF'R - move everything back

This is one case where it is more efficient to temporarily break G2. At the end of the move, everything is restored. This move sequence will do swapping on the bottom side, and at the rear/back. The front/top pair of corners will be left alone. When you are looking for corners that match or mismatch, you will usually find a pair that are not like the rest of the pairs (eg. if all the other pairs match, this pair will mismatch). If you find this odd pair and hold it at the top/front, doing the above sequence will harmonise all the corners.

Now for the edges, there is nothing special here. The edges also lie on circuits, but there are 12 edges and 3 circuits. The edges in the middle layer were already done in step 2 and haven't left their circuit. There are only 8 edges left to deal with, and we must move the white edges to the white sides (front/back), ie. into their circuit, and the green edges to the green sides (left/right). There must be an even number of edges that are facing in the wrong direction. The easiest move to do here is to swap two edges on the green sides for two edges on the white sides. The edges on the greens side must be positioned on the bottom, and the edges on the white sides must be positioned on the top. Then do:

D'

M2 (M is the slice between L and R) - swap!

D

This is self explanatory. Like in step 1, if you only have 2 bad edges, you can make it into 4 in a similar way.

At the end of this step, you should only have two (opposite) colours on each side of the cube.

Step 4.

The cube can now be solved by only using double turns. However, we will break this rule when we see an obvious short cut.

First, using only double turns, solve the corners. First, pick a side, then join two corners on that side and the other two corners for that side will automatically join (if you did step 3 properly). It is then easy to join the two pairs together, and the other 4 corners will be automatically joined (if you did step 3 properly).

To solve the edges, there are two simple moves that are easy to understand:

F2R2 F2R2 F2R2 - Swap two pairs of edges on different circuits. The other pieces return to normal.

F2 M2 F2 M2 - Swap two pairs of edges on the same circuit. The second F2 returns the other pieces back to normal.

Using variations of these, it is possible to solve all the edges.
Variations of the first include:

R (F2R2 F2R2 F2R2) R'
L2B2R (F2R2 F2R2 F2R2) R'B2L2

Variations of the second include:

F2 M F2 M' (the pairs overlap so as to rotate 3 edges)
F (F2 M2 F2 M2) F'

CUBE SOLVED

Ryan

Section 4. Morwen B. Thistlethwaite Original description

THE 45-52 MOVE STRATEGY

Let

$G = \langle L, R, F, B, U, D \rangle$, $G_1 = \langle L, R, F, B, U^2, D^2 \rangle$, $G_2 = \langle L, R, F^2, B^2, U^2, D^2 \rangle$, $G_3 = \langle L^2, R^2, F^2, B^2, U^2, D^2 \rangle$.

The plan is to manoeuvre down through the chain $G = G_0 > G_1 > G_2 > G_3 > 1$. One gets from G_i to G_{i+1} by using moves in G_i only. In its shortest form this strategy would be executed with the help of a computer, in which case I conjecture only 45 moves would be needed, but here I have sacrificed 7 moves in order that there should be no need for a computer. With a few more pages of tables, the figure 52 could be reduced to an intermediate figure of, say, 49. I intend to do this shortly! The indexes of the chain of subgroups are 2048, 1082565, 29400, 663552, but these figures are considerably reduced by considering symmetries. The reader may check that these indexes multiply together to give the order of G . The accompanying tables are broadly classified according to corner positions, and in detail according to edge positions. The words listed actually produce the positions under consideration, so the restoring moves are the inverses of these. In order to be able to use the tables it is necessary to understand the basic characteristics of the groups G_1 , G_2 , G_3 , so the necessary facts are presented below.

Getting into G_1 .

This involves edge pieces only, and is easy, for which reason no tables are given. An edge piece is BAD if in taking it home an odd number of quarter-turns of U and D faces is needed; otherwise it is GOOD (note that badness is well-defined). The reader may quickly work out a rule of thumb for deciding whether a piece is GOOD or BAD. Now quarter-turns of either U or D faces convert BAD pieces to GOOD and vice versa; other moves have no effect. Therefore to make all edge pieces GOOD, move groups of them to U or D face avoiding quarter-turns of U or D, and then cure them by performing a quarter-turn of U or D. For example, if all twelve are BAD, $DBFUR'L'D$ will cure them all!

Getting from G_1 into G_2

What has been achieved so far (although it doesn't look like it) is the correct orientation of edge pieces! In the present stage, the same is accomplished for corners, and also the edge pieces FU, FD, BU, BD are brought into their slice. As is well known, corners do not in general have a natural orientation, but here, roughly speaking, we shall line them all up the same way. More precisely, note that each corner piece has either a L-facet or a R-facet: on completion of this stage each of these facets will lie on either the L or the R face. In fact, the same will be true of the eight edge pieces with L or R facets, in view of the statement above regarding FU, FD, BU, BD. There are 1082565 cases to consider here, this number being the product of 3^7 (total number of corner orientations), and $12C4$ (total number of arrangements of the set $\{FU, FD, BU, BD\}$ amongst the twelve edge positions). Surprisingly, with a certain amount of practice it is possible to get through this stage in at most 17 moves without tables; the same is most certainly not true of the next stage although there are only 29400 cases. However, with a few pages of

tables this figure of 17 may be reduced to 13; with a great deal more computation, it should be possible to reduce it further to 10. To "prove" that 10 moves were sufficient, one would run through all 10-move sequences on the computer, and check that 1082565 different cases resulted. This would take no more than a few hours of computer time, in view of certain short cuts available by considering symmetries. Now to business. The twist of a corner is measured by looking at its L or R facet and observing how this has been rotated in relation to the adjacent L or R face. Note that quarter-turns of F and B faces alter the twist of corners, whereas all other moves in the group G_1 have no effect. Now in at most 4 moves in G_1 try to obtain a position where either the corners on the F face or the corners on the B face have zero twist. Note the twist of the corners on the opposite face and also the position of the edge pieces the edge pieces FU, FD, BU, BD are all in the UD-slice. If this is not possible, get them all in the U-face in at most 4 moves.

FU, FB, BU, BD, and then refer to the appropriate detailed table. The words listed in these tables need to be inverted, as mentioned earlier.

Getting from G_2 into G_3

This is the trickiest stage theoretically, and may be broken down for purposes of clarification into two sub-stages: first get corners into their natural orbits, and second permute the corners within their orbits so as to obtain one of the 96 corner permutations in the squares group (G_3), while at the same time sorting out the edge pieces into their correct slices. The table of initial moves on the first page of Stage 3 tables does part of the first of these sub-stages, and the detailed tables do the rest. After performing the initial moves, the set of corners out of orbit will be one of three possibilities (modulo symmetries): (i) the empty set; (ii) the set $\{1, 8, 2, 7\}$; (iii) the set $\{1, 5\}$. The reader then has to calculate which coset of form $G_3\alpha\beta$ the permutation of corners lies in, where α is one of 1, (14)(68), (24)(58), (12), (14), (24), and β is one of 1, (18)(27), (15). Since some of these cosets are reflections of others, it was not necessary to produce tables for all six possible values of α . The task of reflecting positions if necessary is left to the reader. I apologise for the slightly anomalous numbering of edge positions for this stage; this was due originally to a typing error in a program and has stuck ever since!

Getting from G_3 to home

In this final stage one uses only 180 turns. The order of G_3 is $96 \times 6912 = 663552$. The tables for this stage give words for producing each of the 6912 edge positions with corners fixed. Therefore one must first restore the corners (in at most 4 moves), and then use the tables to restore the edges. Considerable practice is needed to use these tables efficiently, but I have found (after considerable practice) that I can find the desired move in about 2 minutes. One hint is that when faced with three 3-cycles, consider the configuration of the fixed pieces. Also it pays to get to know the different sorts of 4-cycle.

STAGE 2 TABLES FIRST BATCH OF NUMBERS: THE MOVES. 11=L 12=L² 13=L'21=F 22=F² 23=F'31=R 32=R² 33=R'41=B 42=B² 43=B'51=U² 52=D² SECOND BATCH OF NUMBERS: TWISTS OF CORNERS. 1=CLOCKWISE 2=ANTICLOCKWISE 0=NO EFFECT FINAL NUMBER:

NUMBER OF SYMMETRICALLY EQUIVALENT POSITIONS.N.B. FOR INITIAL 1X READ 21 1X.

A DETAILED EXAMPLE OF THE 52-MOVE STRATEGY

$(UFL)_+ (URF)_- (UBR)_+ (ULB)_- (LF)^* (FR)^* (RB)^* (BL)^*$ First decide on a coordinate system for the Cube(i.e. decide which colours are L,R,F,B,U,D); then get it into the above position.Stage

1. There are 4 bad edge-pieces, namely in positions LF,FR,RB,BL. Manoeuvre these to the U-face by $FLR'D^2B^2$.Then the move U corrects them. Summary of Stage 1:- $FLR'D^2B^2U$ (6 moves).Stage

2. The LR-slice edge pieces are now in positions LD,FD,RD,BD. Manoeuvre these to the UD-slice by $F^2D^2LR'F$.Now taking the corner positions in order (as in the diagram in Stage 3 instructions), the respective twists of the pieces in these positions are 0,2,0,2,0,0,1,1. This combination of twists is not given in Stage 2 tables, but a 180° rotation about the LR-axis followed by reflection in the LR-slice transforms this to 2 2 0 0 0 1 0 1, which is in the tables. The move given is $F LFL2F'LF^2BL^2$, and transforming this by the above (involuntary) symmetry gives $B'R'B'R^2BR'B^2F'R^2$. Therefore we perform the inverse of this move, after which L and R faces have L and R colour on them only. Summary of Stage 2:- $F^2D^2LR'FR^2FB^2RB'R^2BRB$ (14 moves)Stage

3. The positions where corners are out of orbit are numbers 1,2,5,8. The preliminary instructions for this stage instruct us to perform $L'U^2$. For the remainder of this stage alter the coordinate system so that the original D-face faces you and the original F-face faces upwards. In this new coordinate system the positions where corners are out of orbit are 1,5. The permutation of corners is (1357)(24). Multiplying this on the right by (15)(24) gives (13)(57) which is a permutation of corners in G_3 . Therefore we must refer to page 7 of the Stage 3 tables. The edge-pieces of the FB-slice are in positions 3,4,5,8. The tables give us $LF^2L'U^2LF^2R^2F^2B^2R$, or $LD^2L'F^2LD^2R^2D^2U^2R$ in the original coordinate system. Perform the inverse of this move. Summary of Stage 3:- $L'U^2R'U^2D^2R^2D^2L'F^2LD^2L'$ (12 moves).Stage

4. The corners are restored by L^2R^2 (original coordinates). Looking at edge-pieces, as no slice is fixed and there is a 2-cycle and a 4-cycle, we refer to page 7 of the Stage 4 tables. The only entries where the correct arrangement of pieces is permuted and the 4-cycle is of the correct type are (143) (1342) (34) and (143) (1243) (34). If we hold the Cube with the original R-face facing the operator, and the original B-face uppermost, we find we have the second of these permutations. Therefore perform the inverse of the move given, i.e. perform $B^2R^2F^2R^2D^2U^2B^2U^2R^2U^2R^2U^2$ in original coords. Summary of Stage 4:- $L^2R^2B^2R^2F^2R^2D^2U^2B^2U^2R^2U^2R^2U^2$ (14 moves). Total number of moves required: $6+14+12+14 = 46$. (45 with cancellation)

M.B.T.

第十六节 降群法简明教程

识破天 整理

概述

这里不介绍此方法背后的原理，只讲解具体实现每一步骤的转动技巧。

降群法分四步走：

- 一.调整 12 个棱块的朝向
- 二.调整上下面颜色为高级色
- 三.调整前后面颜色为中级色
- 四.180 度转动还原

第一步：将棱块朝向翻转正确

这一步可以用所有转动 $\langle U, D, L, R, F, B \rangle$ 。

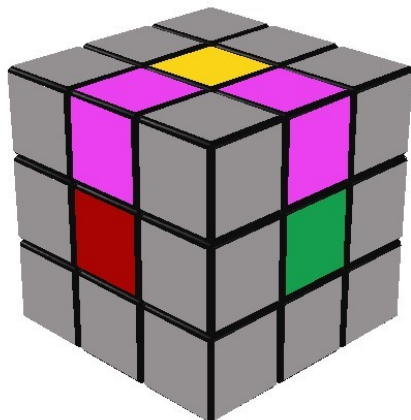
定义：上下面为高级面，前后面为中级面，左右面为低级面。

我的魔方这么放置，上黄下白，左蓝右绿，前红后白，那么，高级面相应的黄白色就定义为高级色，红橙定义为中级色，蓝绿定义为低级色。

判断：一个棱块有两个颜色，处在两个面上。如果它的级别高的颜色处在级别高的面上，它的朝向就是正确的。如果它的级别低的颜色处在级别高的面上，它的朝向就是错的。

只要是涉及棱块朝向对错的判断，如 ZZ 的 EOLine，桥式的 6E4C，盲拧，等等都可以用这个通用的判断方法。

错误棱块调整方法：把四个朝向错的楞块通过 $\langle U, D, L, R, F^2, B^2 \rangle$ 转动移到 F 面(或者 B 面)，做一个 F(或者 B)转动，这四个棱块朝向就都翻转正确了。假如最后还剩下两个棱块方向错误，如下图粉色棱块， FRF' 就调整正确了。



这一步难度在于判断十二个棱块的朝向哪些正确哪些错误。但是可以顺手把底十字做了，就

剩下中上两层八个棱块需要调整方向，判断就大大减少。

第二步：将上下面颜色调整为高级色

也就是说把黄白两种颜色的贴片全部转到上下两个面上。这一步只能使用<U,D,L,R,F2,B2>转动，这几个转动不会改变棱块朝向的正误。

可以分为两个小步：

1. 把棱块的黄白贴片转动到 U 面和 D 面上

通过 L, R 转动，使得顶面有三个棱块是黄白色，底面也有三个棱块是黄白色。把上下面所缺的那两个棱块用 U, D 转动转到 L（或者 R）侧面，剩下两个黄白色棱块必定在中间层，也转到这个侧面，做一个 L（或者 R）转动，UD 面棱块就都调整为黄白色，中间层棱块全部回到中间层。

2. 把八个角块的黄白色贴片转动到 U 面和 D 面上

OLL 21 	$(RUR')U(RU'R')U(RU2R')$	OLL 22 	$RU'^2(R2U')(R2U')(R2U'^2)R$
OLL 23 	$R2-D'-RU2R'-D-RU2R$	OLL 24 	$r(U R') U' (r' F) (R F')$
OLL 25 	$F' r (U R') U' (r' F) R$		
OLL 27 	$(RUR')U (RU2R')$	OLL 26 	$(RU2R') U' (RU'R')$

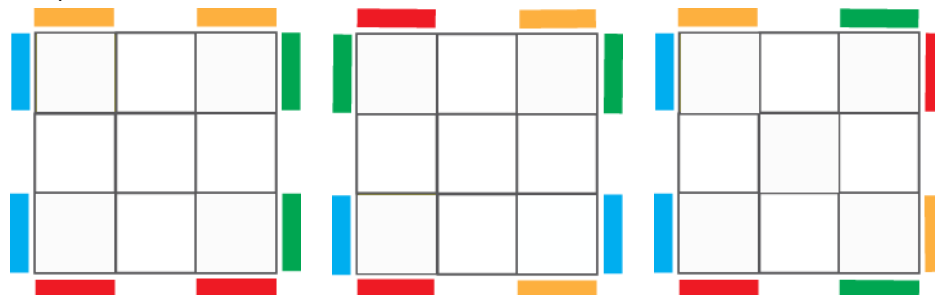
往往四个以上的角块颜色朝向在上一步都自动调整好了，只需用 L2, R2 转动把四个朝向正确的角块转动到同一个面上比如 U 面，这时只需在另一个面上用这七个公式。

第三步：调整前后面颜色为中级色

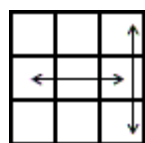
这一步只能用<U,D,L2,R2,F2,B2>转动魔方。严格说来仅仅调整前后面颜色为中级色不充分，有的情形还需要对调一对角块（同时对调一对棱块），才满足降群法第三步的要求。这一步也分为三小步：

1.调整八个角块，使其中级色贴片转动到前后面上

看 U/D 层的四个角块侧面颜色分配。主要是下面三种状态，如图：



第一二种情形中级色都在中级面上，只有第三种情形，有两个中级色在右面，两个低级色在前后面，这种情形只要用 PLL 的一个公式对调右面的两个角块，把中级色转动到前后面上。



R2U'R2DB2L2UL2D'B2

2.把棱块的中级色贴片转动到前后面上

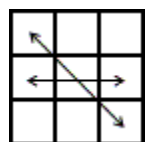
这时，角块位置已经调好了，不能再使用 U，D 转动，以免破坏角块相对位置，或者只能这么用 $U\{U2,D2,L2,R2,F2,B2\}U'$ 。

首先把四个红橙色棱块调到 D 层，把四个蓝绿色棱块调到 U 层。D'-M2-D 是将当前 S-slice 在 D 层的两个楞块放到末态 M-slice 的 U 层，将当前 M-slice 在 U 层的两个楞块放到末态 S-slice 的 D 层。（换一种说法，就是把 DL，DR 两个棱块和 UF，UB 两个棱块位置互换）。

有一种情况是 M-slice 上面只有一个 S-slice 的楞块，这时，用 D'-M2-D，让 M-slice 上有两个 S-slice 的楞块。之后再用上面的方法。

3.将八个角块放到正确相对位置

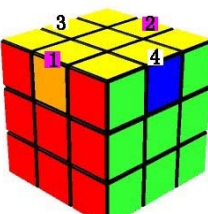
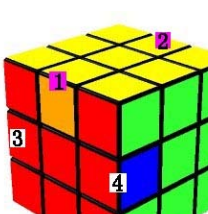
一般情形，要把八个角块放到正确相对位置，需要对调两个对角角块，再简单的 L2，R2 转动即可。



RU'R2'F2U'RF2'R'UF2R2UR'

第四步：180 度转动还原魔方

这一步只能用 $\langle U2,D2,L2,R2,F2,B2 \rangle$ 转动魔方。

Case1		MD2M' 还原 UF 这一个棱块
Case2		M2U2M2 UF<-->UB, DF<-->DB
Case3		U2R2U2R2U2R2 1<-->2, 3<-->4
Case4		M2U2M2-U (M2U2M2) U' 1<-->2, 3<-->4
Case5		E-U2R2U2R2U2R2-E' 1<-->2, 3<-->4

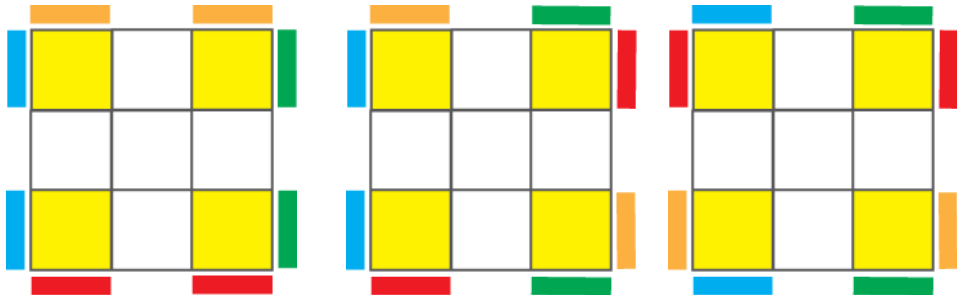
补充说明一下，上述公式中比如第四步的 Case4，用到了 U，U'转动，是为了为了步骤简便（同时保证不破坏魔方状态）。实际上这些操作完全可以照这个方法本身要求的那样只用 { U2,D2,L2,R2,F2,B2}实现。比如 Case4：

M2U2M2U2--y'--U2R2U2R2U2R2---x2y--U2R2U2R2U2R2—变成 E 层三个棱的轮换，再用 F2U2R2U2 F2U2R2U2 轮换这三个棱。

第三步：调整前后面颜色为中级色 另一种玩法：
 这一步也分为两小步：

1. 调整八个角块，使其中级色贴片转动到前后面上

先黄色角块全部调到 U 层，同时白色角块都回到 D 层。这时，看 U/D 层各自的四个角块相对位置是不是正确。比如 U 层角块，有三种状态，U4，U1，U0 如下图：



第一种四个角块侧面颜色完全匹配；第二种不匹配，有且只有左面一个侧面同色；第三种完全不匹配。底面同样判断。

U0 D0	R2F2R2
U1 D0	(R' U R') B2 (R U' R)
U1 D1	(R2 U)(F2 U2 R2)(U R2)
U1 D4	(R U' R) F2 (R' U R) (F2 R2)
U0 D4	(R U' R2)(F2 U' R) F2(R' U F2) (R2 U R')

解释一下我的记法：U1 D1 表示 U 面有一对匹配的，D 面有一对匹配的。先把匹配的边用 U/D 转动转到 B 面上，然后用公式一次即可把八个角块位置调整正确。

2. 把棱块的中级色贴片转动到前后面上

这时，角块位置已经调好了，不能再使用 U，D 转动，以免破坏角块相对位置，或者只能这么用 U{ U2,D2,L2,R2,F2,B2}U'.

首先把四个红橙色棱块调到 D 层，把四个蓝绿色棱块调到 U 层。D'-M2-D 是将当前 S-slice 在 D 层的两个楞块放到末态 M-slice 的 U 层，将当前 M-slice 在 U 层的两个楞块放到末态 S-slice 的 D 层。（换一种说法，就是把 DL，DR 两个棱块和 UF，UB 两个棱块位置互换）。

有一种情况是 M-slice 上面只有一个 S-slice 的楞块，这时，用 D'-M2-D，让 M-lice 上有两个 S-slice 的楞块。之后再用上面的方法。

第十七节 My Corner-first Method

8月29号花八元钱买了个魔方，30号下午开始玩，到今天9月11号，废寝忘食的玩了十二天半。回来照着说明书上的层先法玩了起来，遇到一些写的不清楚的地方求助于 google 查对后终于拼好了。而这上网一查才发现居然有这么多解法，层先，棱先，角先，CFOP, CFCE, Roux, Petrus, Heise, ZZ, ZB, VH, 8355, M2/R2 盲拧，等等十来种。我尽可能的把所有方法都搜到，照虎画猫的每种方法都玩了几次。

这些玩法中，除了 8355 基本不需要公式随手可玩外，其他方法我都嫌公式太多，只是学了它的解法，还没有开始去记公式。在我看来，解魔方本来是个智力游戏，背公式就成了记忆力游戏，公式记熟了就成了动作游戏。普通玩家短期内玩到一到两分钟还原就足够，再往上提速收益曲线就开始陡然平滑，得不偿失了。

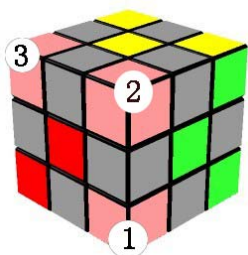
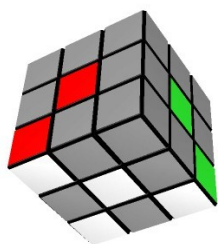
开始玩时，我就试图寻找那种规律性强公式少的解法。规律要比公式更概括，还可以被理解，而复杂的公式只能靠记忆。今天晚上玩着玩着，想到一种解法，公式要尽量少，每个公式还要尽量短，于是修正，查寻 OLL, PLL 公式表，最后清晰成型，写下来大家分享一下。

首先，考虑到任何一个基本转动如 R 会移走某个面如 U 层上的两个角块，但只移走一个楞块。再加上角块有三个面，楞块只有两个面，所以处理角块比处理楞块要复杂。在桥式玩法，F2L, 棱线，角先等等玩法的过程中也体验到这一点。处理楞块比处理角块有更大的自由度，而还原的过程，可以利用的基本转动群是在缩小的，是个自由度减少的过程。综合这两点，如果想找一个简单的解法，合理的考虑就是先解决角块在解决楞块。

按类型算角先法。普通的角先法，解决了 D 层角块之后就会开始对 U 层朝向，再调整位置，需要九个公式。如果一不到位，COLL 的话，公式更多。我的想法是，在解决 D 层角块的同时，用基本转动能不能减少 U 层角块的状态数。实际上，再解决 D 层最后一个角块时至少可以使 U 层有一个角块颜色朝上，这就可以减少角先法两个公式。但这还不够。出于对称性考虑，上下两层角块地位一样，只要使上下两层能够简单复原的角块总数尽量多就足够。肯定能做到的是复原 5 块，底层三块，顶层两块。剩下的三块有粗略的算还有 162 种状态。涉及到三块轮换，两块互换，两个方向扭转或三个扭转。能够解决各种情形的最少的公式是两个，即对换一对角块，同时扭转一对角块。有这两个公式这样就把八个角块全部还原了。当然，还可以用一个公式一对一的对换角块，直到三个角块都归位，同时利用相似变换调整角块朝向。但这个公式至少有 15 个动作不够简洁。剩下还原楞块顺风顺水，最后一步运气的好情形全部还原，但普通情况是剩下两个楞块位置正确，需要翻转，这需要一个简单的翻转公式。这是这个方法的概要，总共只需要 3 个公式。下面来详细介绍。

Step 1 还原底层三个顶层两个角块

下面颜色标记为下白上黄，左蓝右绿，前红后橙。



剩下的三个角块我用粉红色标记。顶层还可以有一个 U 转动自由度，所以随便还原两个都行。此处为了清晰，选择 UBL , UBR 两个角块。这一步略有一点小技巧，先把一个角块方向调好用 F' 或者 R 转动隐藏到底层， U 或 U' 将其邻角放到邻位，再 F 或 R' 把它转入位置。实际上顶层这两个角块选对角选邻角都可以，选对邻角，便于把剩下三个角块放在 F 面上好操作。

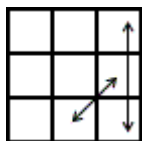
Step 2 还原最后三个角块

最后三个角块颜色朝向和位置可能都不对。但可以通过位置对换归位。再同时扭转两个角块调整朝向。首先把其中那块底块对换到他的正确位置。比如底层 FRD 角块在 2，而 FRU 角块在在 1，用公式 $Alg1$ 对调 12 即可。

Alg1: $R U^2 R' U' R U^2 L' U R' U' L$

它是 PLL 的 J-b permutation.

J Permutation : b



对换是个奇置换，所以不可能只对换两个角块，附带效果是对换了两个楞块，保持所有其他角块不动。

位置对换正确之后调整方向，首先底层 FRD 角块方向对否，不对的话，扭转。顺序是先扭转底块和它的临块，在扭转剩余的两块。即先扭转 12 位置的两块，再扭转 23 位置的两块。当然，要把魔方整体转动成下图公式所示位置。这里就看到选择留顶层临块的好处，这么选使得三个块处于 123 这三个临近位置，总可以使对调或者扭转的那一对处于一个边上，所以各自只用一个最短的公式。

这是我用到的第二个公式 **Alg2**，它是由对易子生成的。我选用最短的公式：

Alg2: (F' U2 F R U2 R') -D- (R U2 R' F' U2 F) -D'

这是一个对易子，它等于 $PQP'Q'$ ，其中 $P = RU'R'URU'R'$ and $Q = D$ ，它是三轮换，(这里体现为贴片的轮换,即两个角块的扭转)这么记更好记。



上面公式逆时针扭转右下角，顺时针扭转左下角块。

把公式中括号里那段取逆， $(R U2 R' F' U2 F) -D- (F' U2 F R U2 R') -D' = Alg2'$

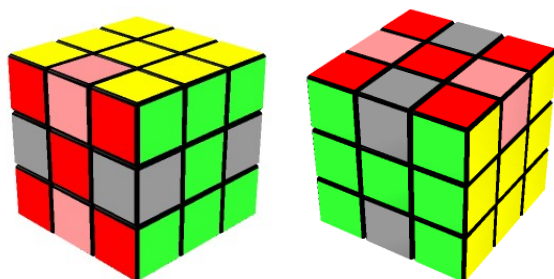
则反方向扭转，即顺时针扭转右下角，逆时针扭转左下角。为方便记忆我们称上图两个角向内翻，用公式 **Alg1**. 反之称向外翻用公式 **Alg2**.

根据 **Twist Parity**，这三个角块方向是两种情形，要么两个一顺时针一逆时针，另一个方向正确，这种情形，把这两。或者是三个都是一个时针方向，假如都是顺时针，这时先用公式 **Alg2** 扭转好底块 1 和它临块 2 这一对，这时 2 变为逆时针，23 一顺一逆在扭转一次即可。

Step 3 还原底层和顶层楞块

还原 8 个角块之后，魔方是完全对称的，第四步选哪两个相对面都行。前三步一直在处理顶面和底面，到这一步对这两个面颜色更熟悉，就选顶面和底面。如果碰巧有一对面上已经有楞块被随机原好了，可以选它们加快速度。

还原楞块是相当容易的，我就不啰嗦了。随后黄色顶面和白色底面各留一个缺口，如图粉红色的两个方块。完成之后把魔方整体翻转，黄色成右面，白色成左面，红色上面，绿色前面。



现在，左右面上的这两个楞块 **LU**，**RU** 必须配成对一起插入这两个缺口。注意这时只转动 **M**

层和 U 层。配对的技巧即是配对这个概念本身：让它们处于 M 层邻位，他们的红色贴片朝向同一个方向，即可插入那两个空位。熟悉桥式玩法的朋友对这一步应该非常熟悉。

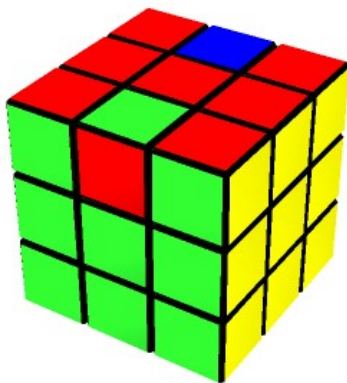
Step 4 还原所有剩余楞块

LU/RU 楞块还原之后，只剩下 M 层上的四个楞块。同样，通过转动 U，M 把它们还原。

最常出现的情况是最后只剩下两个楞块，位置正确，朝向不对。这两个楞块有时可能是 M 上两个临块，有时是 M 层的对角线上的两块。

如果 M 层最后两块处于邻位，用公式 Alg3 即可即可同时翻转他们，同时保持所有其他块不动。 $F'E2F2E'F' - U'2 - FEF2E2F - U2$ 。这个公式用了太多 EF，玩起来不顺手，我利用另一个公式：

Alg3: $(M'UM') U (M'U2M) U (MUM) U2$



假如是两个对角楞块，比如 UF，DB 两块，转动 B2，使 DB 块到 UB 位置，再用公式 Alg2，翻转之后，B2 转回去即可。

第十八节 Blindfold Cubing

盲拧原理:分步还原,每步交换局部的几块保持其他方块位置不动。这么做是为了算法简单,切还原过程中魔方始终保持初始状态(除被还原的局部几块外),以减少记忆量。

五步盲拧法:角块方向→棱块方向→角块位置→棱块位置→奇偶性校验五个步骤还原。有 Grant and Betty's, Richard Carr 等使用。

四步盲拧法:省略了奇偶性校验。奇偶性校验,即最后互换一对角块一对楞块,三循环不适用,用 PLL。

二步法:角块位置和方向→棱块位置和方向→奇偶性校验(有时可省)。由 stefan pochmann 发展。它的升级方法是 Pochmann M2R2 方法。

3OP (3-cycle Orientation Permutation): Stiff hands 发展, Macky 等广泛使用的三轮换方法。也即前面的四步法。

FreeStyle Blindfold, 国内的刁于法,运用三循环的原理一次同时把两块的色向及位置复原,分角块复原及棱块复原两步。

这些方法也可以结合使用,如 Kai Jiptner 的玩法:

Orient Corners

M2 for Edges

3-Cycles for Corners

Section 1. Stefan Pochmann Blindfold Cubing

Invented in 2003, the Pochmann Method was the first method to combine orientation and permutation. Although a cycle method like 3OP, it relies on 2-cycles and can handle everything including the permutation parity using the idea of "a buffer cubie."

基本想法是每次只交换两个块，而不改变所有其他的块。实际上是做不到的。魔方每个动作都是偶置换。

解决方法是，交换两个棱块时，同时交换两个选定的角块。在下一步中交换另两个楞块时再把这两个角块交换回去。这两个角块即为缓冲块。

Stefan 方法同时对位对色。

1.Basic Algorithms:

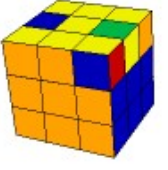
	$(RUR'U')(R'FR^2U'R'U'RU)(R'F')$ PLL Alg.
	$(L'U)(RU')(LU^2)(R'U)(RU^2R')$ PLL Alg.
	$(LU')(R'U)(L'U^2)(RU')(R'U^2R)$ PLL Alg.
	$(RU'R'URUR'F')(RUR'U'R'FR)$ Not PLL Alg. Swap corners, ULB→RDF, same time it swap edges UL→UB.

也可以用其他算法交换块，上面是 Pochmann 用的算法。

2. Set-Up Moves

我们需要交换 AB 两个楞块，作为附带交换了 CD 两个角块。BCD 紧挨着，那么我们把 A 位置称作目标位，B 位置成为缓冲位，BCD 位置合称作缓冲区。注意，说交换是对实体魔方块而言，说目标缓冲等是指位置。要把 B 位上的块移动到目标位 A 上。

假如交换情形不在上述种算法中，做一个相似变换，首先把某个要交换的目标块，移动到目标位置，但不要破坏缓冲区，即剩下的三块的位置。这个相似变换称为 Set-Up Moves。

	(d2L) (RUR'U')(R'FR²U'R'U'RU)(R'F') (L'd2) 想交换 UR->FR,先 d2L 把 FR 转到 UL, 再应用上面的 T-PLL Alg, 之后再 L'd2 把它转回 FR 位。Set-up 动作 d2L 不移动另外交换的三块。
	(d'L') (RUR'U')(R'FR²U'R'U'RU)(R'F') (Ld) 同上, 想交换 UR->FR, set-up 到标准状态时要注意朝向。 Set-up Moves: d'L'
	L d' L - R U R' U' R' F R² U' R' U' R U R' F' - L' d L' 交换 UL->RU, 注意交换这两个方块, 实际上是交换两方向一块交换。

数学解释一下。

交换两个块, 实际上是交换这两个块上的四个贴片。比如第一个算法 T-PLL, 交换 UR->UL, 实际上是交换 UR 上的 U 贴片到 UL 上的 U 贴片, UR 上的 R 贴片到 UL 上的 L 贴片。

记交换的这两个棱块为 E{90}, B{34}, 剩下被交换的两个角块 C{56}, D{78}. 其中数字表示它的两个贴片。E 位置为目标位。假如想利用的变换公式是 $Algs=(AB)(CD)$, 那么就需要一个 Set-up=(AE) 步骤把目标 E 换到 A, 再利用公式。具体到贴片上:

Algs=[(13) (24)] (57) (68)

Set-up=S=(19)(20)

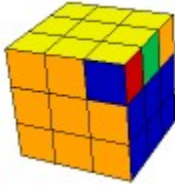
S*Algs*S'=[(93) (04)] (57) (68) [(11)(22)]

效果就是交换了 E 和 B, C 和 D, A 不动。A 是 Set-up 和 Algs 唯一共有的块。从上更清晰的明了相似变换起的作用。

Section 2. Joel Van Noort's Explanations

1. Shooting a piece to UL

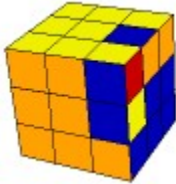
同上面 Stenfan Porchmann。



Shooting UR to UL, at the same time swap URF and URB.

[shoot to UL] R U R' U' R' F R2 U' R' U' R U R' F'

2.Set-up,shooting UR to RF



[shoot to RF]d' L' - R U R' U' R' F R2 U' R' U' R U R' F' - L d

3. 3-Cycles

上面我们交换两个楞块（附带交换了两个角块），假如我们想轮换三个楞块该怎么做呢？

假设我还是用上面的公式 Algs，BCD 是缓冲区，要轮换的楞块是（BEF）。

1) 先用 **set-up1**，把 E 块带到 A 位置，用公式 **Algs** 交换 BE 块，再 **anti-set-up1**，结果是 B 块移到 E 位置，E 块移到 B 位置，A 块仍在 A 位置，CD 块被交换了一次。

2) 现在要做的是把 F 块用 **set-up2** 移到 A 位置，利用 **Algs** 交换 E 块 F 块，再 **anti-Set-up2**，结果是，F 块 E 块交换，A 块没变，CD 块又被交换了一次回到初始状态。

总的结果是仅仅轮换了（BEF）三个块，其他所有块不动！

上述例子是个三轮换，假如我们做四轮换（BEFG），则 CD 块又被交换一次，结果是（BEFG）（CD）。

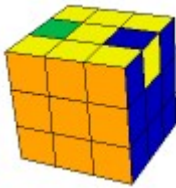
4.Cycles Product 循环乘积

上面处理的是单个循环。比较交换（BE），三循环（BEF）。假如我想用一个公式处理循环的乘积比如（BE）（FGH）该怎么办？把这个循环变成（BE）（EFGH）（EF）即可。

5.Flip Two Edges

楞块的状态可以分解为轮换之积。这个轮换实际上是楞块上贴片的轮换。我们上面的方法是轮换不同的楞块。还有一种情况是，还剩两个楞块，他们位置正确，但颜色朝向需要翻转，即一个楞块的两个贴片需要轮换。实际上，因为贴片的轮换都是四轮换，我们最后总是有偶

数个楞块（即贴片数是 4 的倍数）需要翻转。



如图，想分别翻转 UL, UR 两个楞块。如果还利用上面的方法的话，可以分两步实现，先交换这两个楞块， $R U R' U' R' F R^2 U' R' U' R U R' F'$

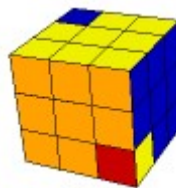
再一次交换这两个楞块，同时对好颜色朝向： $L d' L - R U R' U' R' F R^2 U' R' U' R U R' F' - L' d L'$ 。

用置换群语言来说： $(12)(34) = (13)(24) * (32)(14)$ 。

当然还有简单的办法做到这个，比如 $M' U M' U M' U M' U^2 M' U M' U M' U M'$ 。或者还可以用 Commutator 做到。

6. Corners

上面分析了轮换（交换）楞块的情况。因为要保持 **set-up** 和缓冲区没有交叠，必须用到 **T**, **J permutation**。对于交换角块来说更容易，只需要一个公式即可交换所有的角块。



The position I shoot from is the LBU position. I shoot it to DFR, with one algorithm. It has one side-effect, it swaps UL and UB edges.

把 LBU 块映射到 DFR，同时交换了 UL 和 UB 两个楞块。因为所有角块都可以不动缓冲区的移到 DFR 位置，故只需要一个算法即可。

[shooting to DFR] $R U' R' U' R U R' F' R U R' U' R' F R$

PLL 的 Y-Permutation 就是利用这个公式。

假如角块在正确位置，需要扭转方向。道理同上面的楞块，利用两次公式，先换位，第二次换位是调对方向。当然，还可以用其他方法，例如：

- $R' U' R U' R' U^2 R z U R U' R U R^2 U' z'$
- $R U R' U R U^2 (R U^2 R' U' R U' R') U^2 R' U^2$
- Or, you can use a commutator twat twists corners.

7. Joel Van Noort's Algorithms:

1).For Edges

Joel Van Noot 利用 PLL 的 T-Permutation 和 J-Permutation.缓冲区是 UFR, UR, UBR。

Shoot UR to some other position.

The Basic Algorithms

Name	Shoots to	Algorithm
T1	[shoot to UL]	R U' R' U' R' F R2 U' R' U' R U' F'
T2	[shoot to LU]	x' R2 U' R' U x R' F' U' F R U' U'
J1	[shoot to UF]	R U' F' R U' R' U' R' F R2 U' R' U'
J2	[shoot to UB]	R' U2 R U' R' U2 L U' R U' L'

The following table shows you how to shoot UR to every positions:

Shoot to	Setup moves	Algorithm	Undo setup moves
[shoot to UF]		J1	
[shoot to UL]		T1	
[shoot to UB]		J2	
[shoot to FU]	I'	J2	I
[shoot to LU]		T2	
[shoot to BU]	I	J1	I'
[shoot to FR]	d2 L	T1	L' d2
[shoot to FL]	L'	T1	L
[shoot to LF]	L'	T2	L
[shoot to LB]	L	T2	L'
[shoot to BL]	L	T1	L'
[shoot to BR]	d2 L'	T1	L d2
[shoot to RB]	d L	T1	L' d'
[shoot to RF]	d' L'	T1	L d
[shoot to FD]	I'	J1	I
[shoot to LD]	L2	T2	L2
[shoot to BD]	I	J2	I'
[shoot to RD]	D' I'	J1	I D
[shoot to DF]	I2	J2	I2
[shoot to DL]	L2	T1	L2
[shoot to DB]	I2	J1	I2
[shoot to DR]	S'	T1	S

2). For Corners

The position I shoot from is the LBU position. I shoot it to DFR, with one algorithm. It has one side-effect, it swaps UL and UB edges.

Algorithm [shooting to DFR] R U' R' U' R U' R' F' R U' R' U' R' F R

Shoot to	Setup moves	Undo setup moves
[shoot to URF]	R2 D'	D R2
[shoot to RFU]	F	F'
[shoot to FUR]	R'	R
[shoot to UFL]	F2	F2
[shoot to FLU]	F R'	R F'
[shoot to LUF]	F' D	D' F
[shoot to UBR]	R2	R2
[shoot to BRU]	R D'	D R'
[shoot to RUB]	R' F	F' R
[shoot to DFR]		
[shoot to FRD]	F' R'	R F
[shoot to RDF]	R F	F' R'
[shoot to DLF]	D	D'
[shoot to LFD]	F'	F
[shoot to FDL]	F2 R'	R F2
[shoot to DBL]	D2	D2
[shoot to BLD]	D F'	F D'
[shoot to LDB]	D' R	R' D
[shoot to DRB]	D'	D
[shoot to RBD]	R2 F	F' R2
[shoot to BDR]	R	R'

8.提速方案

最常遇到的三轮换，四轮换，等等，而不是对换。上述方案解决方法是把轮换分解为对换。这样用于步骤很多。所以，用一些三轮换加快速度：

(setup) first target at	(setup) second target at	Algorithm
UL	UF	R U' R U R U' R' U' R2
UL	BU	M' U M U2 M' U M
FD	LD	M' U M D M' U' M D'
DR	DF	y M' U2 M D M' U2 M D' y'
FL	BR	L' U2 L E2 L' U2 L E2
FR	UB	R' U' R' U' R' U R U R U

.....

（注明，三轮换的其实位置还是 UR）

Section 3. M2/R2 blindcubing methods

Invented by Stefan Pochmann in 2006, this uses a very elegant idea and is a vast improvement over his original method. M2 can be used together with with 3OP corners.

1.Edges

DF 做缓冲位，UB 做目标位。M 层做缓冲区。公式是 M2. UB 就是目标位。附带效果是 M 层其它块也被移动了。显然，做两次 M 层就复位。

2.Corners

同楞块，这次用 R 层。DFR 做缓冲位，UBR 做目标位。

Target	Algorithm
UB BU	M2 M2
FR DR BR UR	U R U' M2 U R' U' U R2 U' M2 U R2 U' U R' U' M2 U R U' R' U R U' M2 U R' U' R
FL DL BL UL	U' L' U M2 U' L U U' L2 U M2 U' L2 U U' L U M2 U' L' U L U' L' U M2 U' L U L'
RU RF RD RB	x' U' R U M2 U' R' U x x' U' R2 U M2 U' R2 U x x' U' R' U M2 U' R U x l U' R' U M2 U' R U l'
LU LF LD LB	x' U L' U' M2 U L U' x x' U L2' U' M2 U L2 U' x x' U L U' M2 U L' U' x r' U L U' M2 U L' U' r
DB BD UF FU	M U2 M U2 M U2 M U2 U2 M' U2 M' U2 M' U2 M'

Target	Algorithm
UBR BRU RUB	R2 R2 R2
BLD ULB DLF FLU	U' L U R2 U' L' U L' U' L U R2 U' L' U L U' L2 U R2 U' L2 U R' U L U' R2 U L' U' R
LBU LUF LFD LDB	U' L' U R2 U' L U L' U' L' U R2 U' L U L L2' U' L' U R2 U' L U L2' L U' L' U R2 U' L U L'
BUL DBL FDL UFL	U' F' L2' F U R2 U' F' L2' F U L U' F' L2' F U R2 U' F' L2' F U L' d' L U' L' U2 F2 U2 L U L' d y' (R' U2 R U2') (l U2) (l' U2) (R' U2 R) y
RBD BDR DRB	(R' U R2 U' R' F' R U R2 U' R' F) (R U R' D r2 U') (R U r2' U') (D' R) (R2 U' R' F') (r U R2' U') (r' F) (R' U)
RFU URF FUR	(F' R U R2 U' R' F R U R2' U' R) U' (R F' r U R2 U') (r' F R U R2') (R2' U' r x) (l2' U L U') (R' U) (L' U' L') (R' U)

过程中有几点需要知道：当 DF 缓冲块上是正确的块时，可以把它 shoot 到任意一个错误楞块处，

再接着循环。如果不是盲拧，还可以随意转动 M ，找到一个新的块当缓冲，以它开始重做循环。最后，只剩下 M 层上棱块位置需要调整。结合 $U2MU2$ ，可以削减到互换两个棱块。

Section 4 Macky's 3OP

Also called the 4-step 3-cycle method, this was the method of choice for top blindfold cubers until 2006. Although M2 has begun to replace 3OP edges, 3OP corners is still widely used by top cubers.

Overview

Whatever the method, blindfolded solving is very different from normal cubing. While sighted methods aim for fewer moves and affect many pieces at each step, blindfolded methods use a limited number of basic algorithms that move very few pieces. This makes it possible to keep track of the state of the puzzle while blindfolded. In cycle methods, the solver memorizes the necessary operations left rather than the puzzle's state. As a result, the amount needed to be memorized decreases until there is nothing to remember at all, at which point the puzzle is solved.

Rubik's Cube has four nearly independent parts: corner orientation (CO), edge orientation (EO), corner permutation (CP), and edge permutation (EP). Since a Rubik's Cube has 20 cubies, all the necessary information can be memorized as 40 numbers. The actual memory burden can be significantly eased by memorizing visually and is comparable to two 10-digit phone numbers.

3OP solves the cube one part at a time, starting with the two orientation steps. Each piece is first oriented (flipped or rotated) in place, meaning without changing the permutation. Then, each piece is permuted (moved) to its correct spot, now without disturbing the already-corrected orientation. Each of the four parts is solved independently, except possibly to correct the permutation parity involving two corners and two edges. Each step is in turn divided into smaller tasks--in CP and EP, into cycles--each of which is handled by judiciously applying an algorithm. Because everything can be broken into a small number of tasks, 3OP requires only a handful of algorithms.

Orientation

1. Edge orientation

We define "correct" orientation of an edge to be the one that it can reach from the solved state within the (UDF2B2RL) group*, i.e. without quarter turns on F and B faces. The other orientations are "incorrect."

In U/D layer

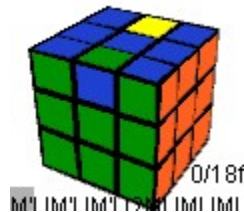
1. If the piece has a U/D color, correct if this is on U/D, incorrect if on F/B/R/L.
2. Otherwise, correct if R/L color is on F/B/R/L, incorrect if on U/D.

In the middle layer

3. If the piece has a U/D color, correct if this is on F/B, incorrect if on R/L.

4. Otherwise, look at either one of the two stickers and the adjacent center. If these two colors are same or on opposite sides of the cube, correct. If not, incorrect.

Once we know how to determine the orientation, this is the easiest step in the 3-cycle method.. Any solvable configuration of the cube has an even number of incorrectly oriented edges(For the reason that the permutation of edge's stickers is even permutaiton). We can therefore proceed by flipping two of these edges at a time, which automatically corrects the orientation, with the following:



Algorithms : $M'UM'UM'U2MUMUMU2$

这是一个很有用的翻转楞块公式。这里也可以用 $F'E2F2E'F' - U'2 - FEF2E2F - U2$ 翻转这两个棱块。

2.Orient Corners

Corner orientation is slightly trickier because there are three possible orientations for each corner: correct, clockwise (hereafter "cw"), and counter-clockwise (hereafter "ccw"). A corner is correctly oriented when its U/D-colored sticker is on U or D.

We first explain how to solve a cw/ccw pair when both corners are on the U layer. We introduce the monoflip:

Monoflip $A=R'D'RDR'D'R$

Monoflip $A'=R'DRD'R'DR$

A' is the inverse of A . Note that A rotates corner 2 ccw and leaves all other U layer pieces intact. A' has a similar effect but twists 2 cw.

Commutator: $U'AUA' = U'-R'D'RDR'D'R-U-R'DRD'R'DR$.

Twist two corners:



Additional algorithms

Although corner orientation can be handled quite efficiently using commutators, there are slightly faster algorithms for special cases. You may wish to learn the following algorithms once you successfully solve the corner orientation blindfolded with commutators.

Algorithms

(43)(21): $RUR'URU'R'URU^2R'-R^2U'R'U'RURURU'R$ (OLL - PLL)

(43)(12): $FRUR'U'RUR'U'F'-RUR'U'RwR'URU'R'w$ (2 OLL's)

(234 cw): $RU-R^2U'R^2U'-RUR'U'-RURU-RU'R'U$



Permutation

Permutation is where the pieces need to go. Our goal is to move all pieces to their correct spot while preserving the orientation, which should already be solved. Like orientation, permutation is also divided into corners and edges; however, each scramble has a 50% chance of having a permutation parity, in which case we need to transpose a pair of edges and a pair of corners simultaneously. The same principle of set-up moves apply here, but with added restrictions to preserve the orientation.

Cycle Method

The permutation method explained here is known as the cycle method and is used for the corners as well as the edges. This is the defining difference between cycle methods and the so-called piece-by-piece method. It is essential that you completely understand the material in this section; solving along cycles is the single most important concept to grasp in any cycle method, including this one.

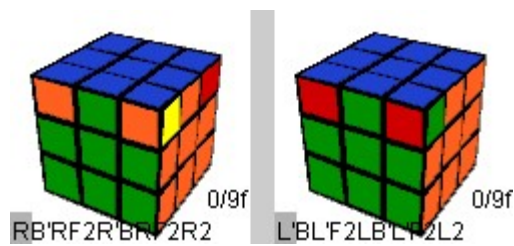
Mathematically inclined readers will recall that every permutation can be uniquely decomposed into a product of disjoint cycles (up to order of the cycles). In a more ordinary language, we can rewrite every configuration of, say, the corners, into a series of permutations in which pieces are cycled. Such as $(1\ 5\ 8)(2\ 6)(4\ 12\ 11\ 7)(9\ 10)$.

3. Corner Permutation

Algorithms

CP(123): $RB'RF^2R'BRF^2R^2$

CP(214): $L'BL'F^2LB'L'F^2L^2$



4.Edge Permutation

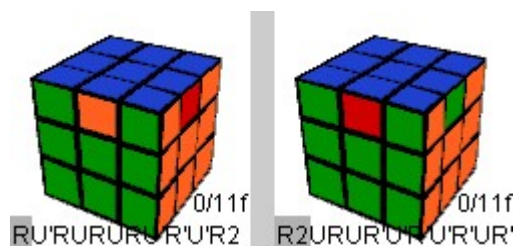
For many people, edge permutation is the hardest part of the 3-cycle method because it involves twelve pieces, more than the number of corners. However, the exact same approach used for corners also applies here; we will still use 3-cycles to reduce the cycles one after another. The only difference is that the set-up moves must now stay within the (UDF2B2RL) group, meaning no F/B single turns, to preserve the orientation.* The increased freedom from the set-up moves for corners also means that we need to be more careful to remember the order of turns correctly. For example, we may be able to use either B2R' or R'B2. Making some rules for the set-up moves, such as performing, whenever possible, U/D first, R/L next, and finally F2/B2, can be helpful in avoiding errors.

As with the corners, it is useful to know the 3-cycle in both directions:

Algorithms

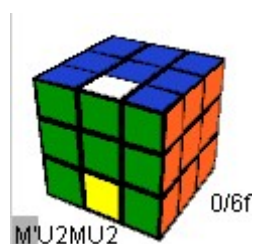
EP(421): RU'RURURU'R'U'R2

EP(241): R2URUR'U'R'U'R'UR'



Algorithms

EP(193): M'U2MU2



5.Permutation Parity

50% of solves will have a permutation parity, meaning an odd permutation of edges and, consequently, an odd permutation of the corners. Blindfold cubers have not reached a consensus on how best to deal with the permutation parity. Perhaps the easiest method is to solve the corners using T permutation and the four edges using H permutation, which we already saw. In these two parts, the set-up moves must obey the same restrictions used for corner and for edge permutation, respectively.

Algorithms

T=CP(23) EP(24): RUR'U'R'FR2U'R'U'RUR'F'

H=EP(13)(24): RLU2R'L'F'B'U2FB

For many people, permutation parity is the hardest part of blindfold cubing. One way to avoid parities altogether is to determine the parity during inspection from the corners (corners are usually easier since there are fewer pieces). Parity is even (no parity fix necessary) if and only if the number of cycles of even length is even. If there is a parity, we can perform U at the beginning of the solve to change this (a 4-cycle is an odd permutation). However, since we cannot make any move during inspection, we must memorize the permutation after an imaginary U.

Summary

Memorization

Memorization of the four parts can be done in any order. Here, we will discuss them in the following order: EP, CP, EO, CO. My reason for using this order is explained in the next section, IV. Memorization.

第十九节 Commutators

Blindfold Cubing 利用了两对交换。 $(AB)(CD)$ 。做相似变换, $(AE)(AB)(CD)(AE)$ 实现交换 $(EB)(CD)$ 。

这一节怎么利用 commutator 生成三轮换 (ABC) , 进而可利用相似变换 $(AE)(ABC)(AE)$ 把这个特定的三轮换变成任意的三轮换 (EBC) 。还可以利用三轮换扭转一对方块。

1. Definition of a commutator

If P and Q are algorithms, the commutator of P and Q is: $PQP^{-1}Q^{-1}$. So first execute P , then execute Q , then 'undo' P and 'undo' Q .

用置换来看对易子。

For example: $P = (123456789)$, $Q = (9\ 10\ 11\ 12\ 13\ 14\ 15\ 16\ 17)$, In this case:
 $PQP^{-1}Q^{-1} = (123456789)(9\ 10\ 11\ 12\ 13\ 14\ 15\ 16\ 17)(987654321)(17\ 16\ 15\ 14\ 13\ 12\ 11\ 10\ 9)$
 $= (9\ 17\ 8)$

一般来说, 如果 P , Q 只有一个公共元素, 那么 $[P, Q]$ 对易子就生成一个三轮换。

P , Q 有两个公共元素, 他们的对易子是两个对换:

$P=(12345), Q=(45678)$

$[P, Q]=(38)(45)$

有一种例外, PQ 有两个公共元素, 而 P 或 Q 只有两个元素, 它们的对易子是个三轮换:

$P=(45), Q=(45678)$, 则有 $[P, Q]=(458)$

PQ 有三个公共元素, 他们对易子是两个对换:

$P=(123456), Q=(456789)$

$[P, Q]=(39)(56)$

这里, 也有一种例外, PQ 有三个公共元素, 而 P 或 Q 只有三个元素, 则他们对易子是个三轮换: $P=(456), Q=(456789)$, 则有 $[P, Q]=(569)$

PQ 有四个公共元素, 他们的对易子仍是两个对换之积:

$P=(1234567), Q=(4567890)$

$[P, Q]=(30)(67)$

同样, 也有一种例外, PQ 有四个公共元素, 而 P 或 Q 只有四个元素, 它们对易子是个三轮换: $P=(4567), Q=(4567890)$, 则有 $[P, Q]=(670)$

用在魔方上, 假如 P 操作的结果是一些位置的置换, Q 操作是另外一些位置的指环, PQ 只有一个公共位置, 那么 $[P, Q]$ 就是一个三个块的轮换。假如 PQ 有两个以上公共位置, 他们的对易子是两个对换的乘积。这可以用来生成两对棱块 (或两对角块) 的对换, 比如 PLL

中的某些情形。我们下面主要利用对易子进行三轮换。

最基本的两个三轮换，角块三轮换和楞块三轮换可以用对易子生成。

角块三轮换(DFR,UFR,UFL): $P=R'D'R, Q=U'$, $[P, Q] = (DFR, UFR, UFL)$.

其中, P, Q 操作只有一个公共角块 UFR, $R'D'R$ 操作($..., DFR, UFR, ...$), U' 操作($..., UFL, UFR, ...$), 所以 $[P, Q] = (DFR, UFR, UFL)$ 。

棱块三轮换 (UB, UF, FD): $P=U'SU, Q=F2, [P, Q] = (UB, UF, FD)$.

其中, $U'SU$ 操作是置换($..., UB, UF, ...$), $F2$ 操作是置换($..., DF, UF, ...$), 它们只有一个公共棱块 UF, 所以 $[P, Q] = (UB, UF, FD)$ 。

这里, 楞块三轮换也可以用有两个公共元素的特殊情形来实现。

$P=M', Q=U2, [P, Q] = (UB, UF, FD)$

其中, M' 操作是置换($..., FD, UF, UB, ...$), $U2$ 操作是(UF, UB)($...$), 它们有两个公共楞块 UF, UB, 而且 $U2$ 操作是 UF, UB 的对换 (乘以其他独立的置换), 所以 $[P, Q] = (UB, UF, FD)$ 。

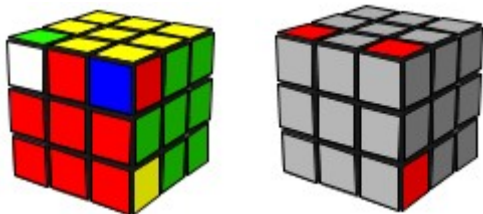
2.Basic Corner 3-Cycles (8 move)

2-a F 面上逆时针三轮换 RFU->LFU->RFD

$P = R'D'R$ and $Q = U'$

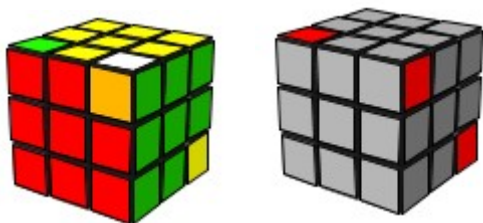
$R'D'R U' R'DR U$

对易子是 F Slice 上逆时针三轮换, RFU->LFU->RFD, 这三个块的三轮换, 实际上是三个块上的贴片之间的 3 个三轮换。



2-b 三轮换

$P = ULU'$ and $Q = R2$, $ULU' R2 UL'U' R2$ 使下面三个角轮换:

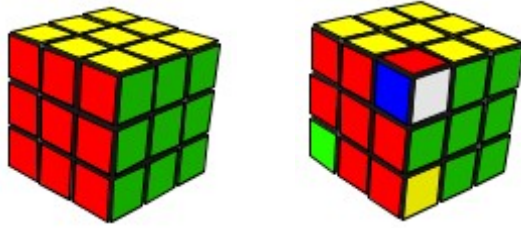


2-c F 面上顺时针三轮换 RFU -> DFR -> DLF

$P = RUR', Q = D, P^{-1} = RU'R', Q^{-1} = D'$

$PQP^{-1}Q^{-1} = RUR' D RU'R' D'$

RFU -> DFR -> DLF, 三个角块轮换时, 这三个角块上的三组个面也在做轮换。



注意，例子中 P 本质上都是 RUR' , $RU2R'$ or $RU'R'$ 这样的操作。任何角三轮换总可以在 12 步以内完成，一般 10 步就可以。上面几个例子都是八步，实现三轮换。如果不考虑角的朝向，任何三个角都可以用上面前两种基本三轮换实现。假如考虑角块轮换后朝向也是正确的，有的三个角不能通过上面基本轮换做到，需要 **set-up**，使它们可以应用基本三轮换。

3. Twist Two Corners

对易子可以用来同时扭转两个块的朝向。

$P = RU'R'URU'R'$ and $Q = D$. P 扭转逆时针 D 层上的 RFD 角，不改变 D 层上任何其他方块。 $[P,Q]$ 扭转 FDL, FDR 两个角，一个顺时针一个反时针。假如想顺时针扭转 RFD，逆时针扭转 LDF 用 P' 即可 $[P',Q]$.



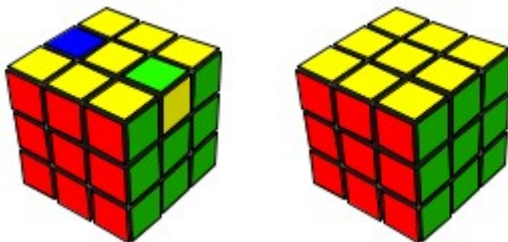
如果取 $Q = D'$ or $Q = D2$ ，那么可以我们扭转底面上其他两个角。

还可以使用 $P = F' U2 F R U2 R'$ 扭转 RFD 角, 再取 $Q=D$ 。注意这是 8355 方法里面扭转角块用到的方法。

4. Flip Two Edges

$P = RE2R2ER$ and $Q = U2$. 可以用它翻转两个楞块。做个相似变换就可以翻转任意一对楞块。

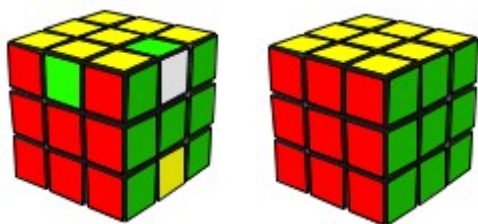
$RE2R2ER - U2 - R'E'R2E2R' - U2$



5. Edge 3-Cycles

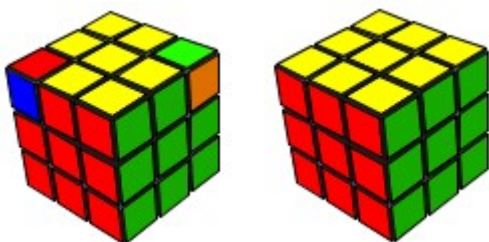
$P = MD'M'$ and $Q = U$.

$MD'M' - U - MDM' - U'$



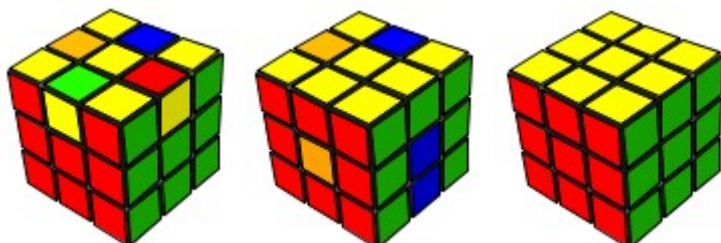
6. 同时扭转两个角两次

$P = F'RD2R'F$ and $Q = U2$. P will swap the ULF and URB pieces, but it will also twist them.
 $F'RD2R'F - U2 - F'RD2R'F - U2$ 则扭转它们两次。



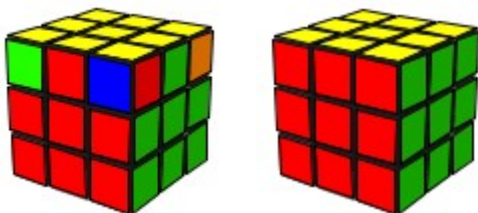
7. 交换同一个面上两对棱

$P = F'R E2 R'F$ and $Q = U2$. 其中 P 交换两个棱 UF , UR (第二个图)。
 $F'R E2 R'F U2 F'R E2 R'F U2$, 交换 U 面上的四个棱。 $UF \leftrightarrow UR, UL \leftrightarrow UB$.



8. 同时交换一个面上两对角

$P = RU'L D2 L'UR'$ and $Q = U2$. You can also try $Q = U$ or $Q = U'$. 其中 P 交换 LUB 和 RUB 两个角。
 $RU'L D2 L'UR' U2 RU'L D2 L'UR' U2$, 交换两对角 $LUB \leftrightarrow RUB, LUF \leftrightarrow RUF$.



其他魔方，比如 **SQ**，星形 **SQ**，**Skewb** 等等，都可以用对易子构造出少数几个块的轮换/置换。

第二十章 Mathematics of Cube

通过运用电脑，Tomas Rokicki 于 2008 年宣布证明了任何魔方可以在 25 步以内解开。而随后，这一结果改进为 22 步。

1. Basic laws

不可能在保持其他方块不动情况下只交换两个棱块或两个角块。

Proof: 转动任何一个面，是四个角块的轮换和四个棱块的轮换，这两个四轮换都是奇置换，其和是偶置换。转动中间层（或者整体转动魔方），是四个中层棱的轮换和四个中心块的轮换，这总体也是个偶置换。单独交换一对棱块或角块是奇置换。故不可以。

不可能保持其他方块不动情况下，只翻转一个棱块。

Proof: 任何一个基本转动，都是四个棱块的四轮换，是棱块的八个贴片的两两四轮换，所以是棱块贴片的偶置换。而只翻转一个棱块是这个棱块两贴片的奇置换。

不可能在不改变其他角块朝向时只扭转一个角块的方向。（Twist Parity）

对魔方的可能状态只有这上面三个限制。

魔方总的状态数为

$$(8! \cdot 3^8)(12! \cdot 2^{12}) / (3 \cdot 2 \cdot 2) = 43,252,003,274,489,856,000$$

其中 8 个角色块全排列 $8!$ ，而每个角色块又有 3 种朝向，所以是 $8! \cdot 3^8$ ，12 个棱色块全排列每个有 2 种朝向是 $12! \cdot 2^{12}$ 。排好 7 个角块后，最后一个角块方向就定了，不能扭转，所以除以 3。同理，排好 11 个棱块之后最后一个棱块方向不能翻转，除以 2。另外所有转动都是偶置换，要除掉这种排列中一半的奇置换。

2. Permutation of Cube

魔方的转动是贴片的若干个四轮换。

任何轮换都可分解为对换的乘积（这里都是向左乘，即左边的操作在先）：

$$(12)(23)(34) = (4321)$$

利用上面的性质可以做特定的对换：

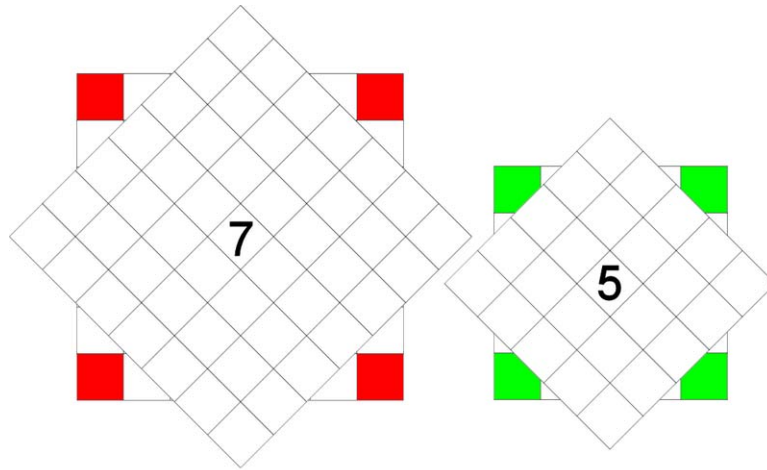
$$(12) = (12)(23)(34) * (34)(23) = (4321)(234)$$

也就是说对换也可以写成轮换的乘积。

这个性质很有用，可以交换特定的几个贴片。

3. 立方体外形

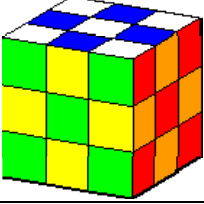
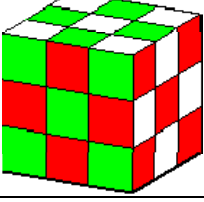
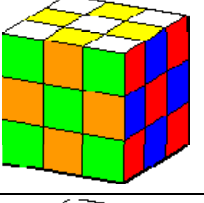
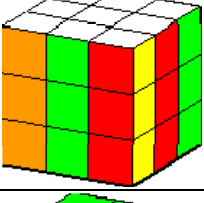
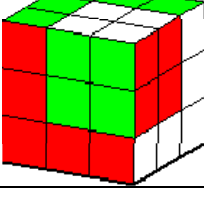
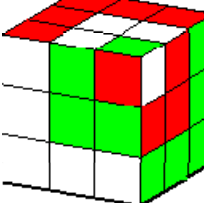
对于立方体魔方，中心到角块的卡脚的距离为 $\frac{1}{n} \times \frac{n-2}{2} \times \sqrt{2}$, 中心到面的距离为 $1/2$ ，为了保证卡脚再转动过程中始终在面内，必须有 $\frac{1}{n} \times \frac{n-2}{2} \times \sqrt{2} < \frac{1}{2}$ ，也就是 $n < 6.8$. 所以七阶以及七阶以上的魔方不可能设计为立方体的外形。

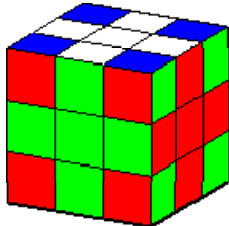
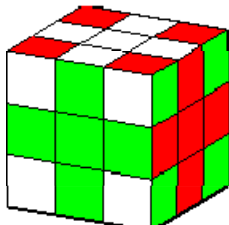
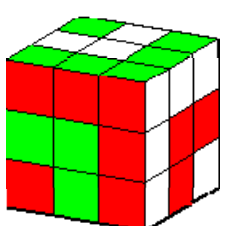
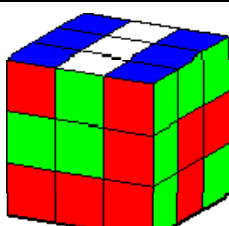
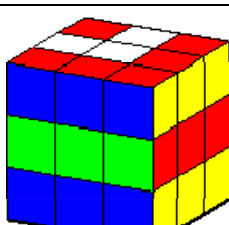
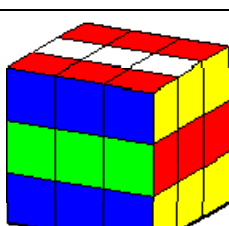
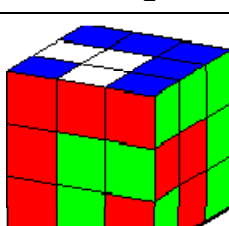


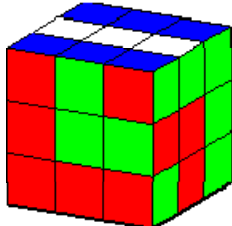
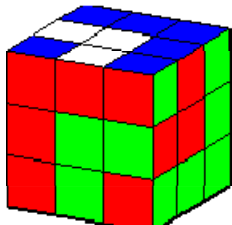
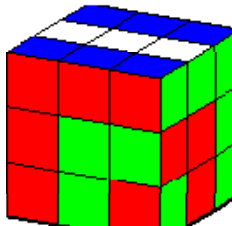
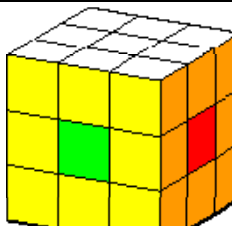
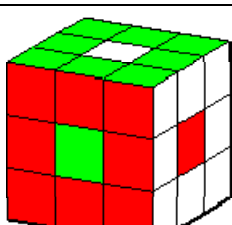
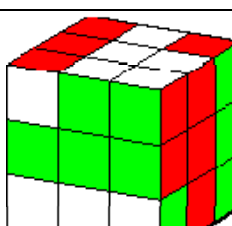
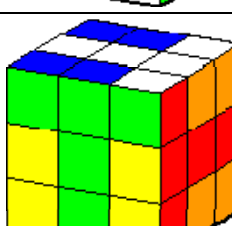
第二十一节 Cube Patterns

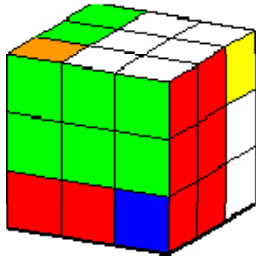
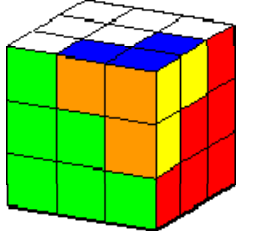
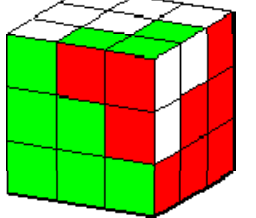
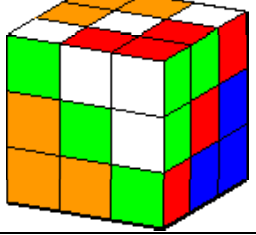
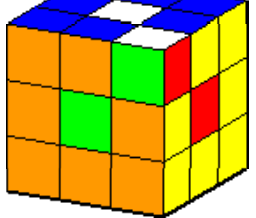
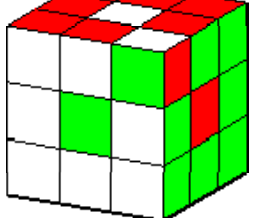
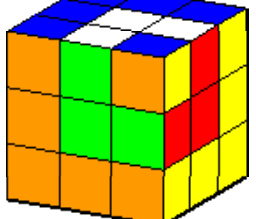
Here are some pretty patterns on Rubik's cube.

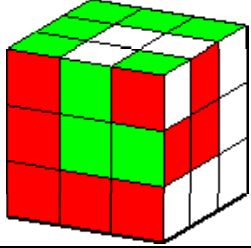
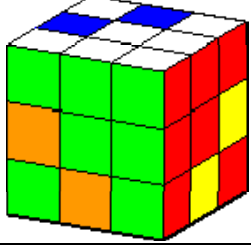
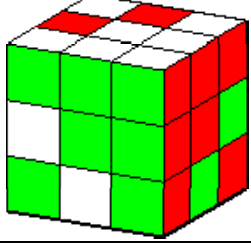
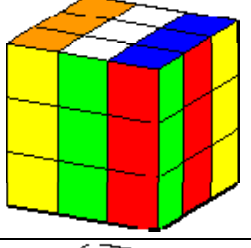
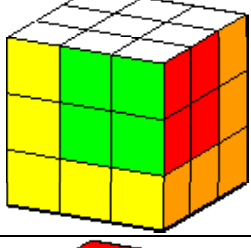
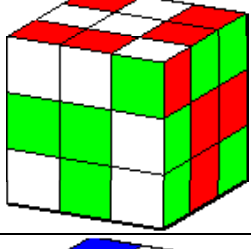
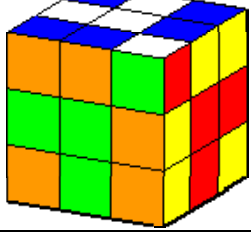
Michael Reid's Patterns

	Pons asinorum $F^2 B^2 R^2 L^2 U^2 D^2$
	Checkerboards of order 3 $F B^2 R' D^2 B R U D' R L' D' F' R^2 D F^2 B'$
	Checkerboards of order 6 $R' D' F' D L F U^2 B' L U D' R' D' L F L^2 U F'$ $R^2 L^2 U B L^2 D' F B^2 R L' F' B R D F^2 L' U'$
	Stripes $F U F R L^2 B D' R D^2 L D' B R^2 L F U F$
	Cube in a cube $F L F U' R U F^2 L^2 U' L' B D' B' L^2 U$
	Cube in a cube in a cube $U' L' U' F' R^2 B' R F U B^2 U B' L U' F U R F'$ $F' U B' R' U F^2 U^2 F' U' F U^2 D B' D' R^2 B^2 U'$

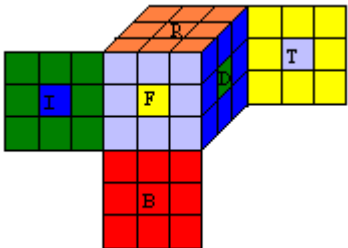
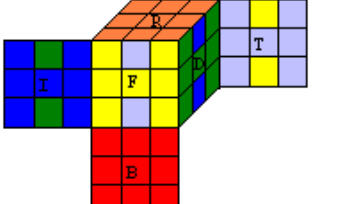
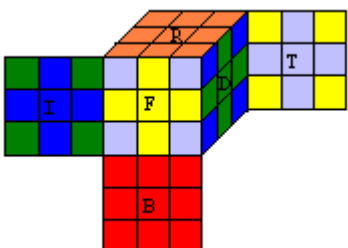
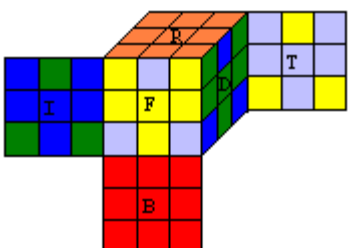
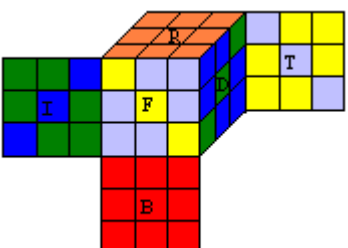
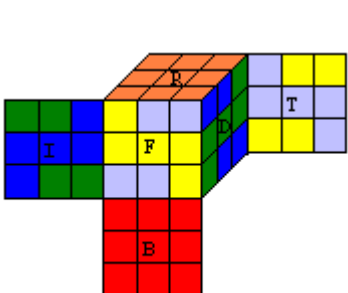
	Christman's cross $U F B' L^2 U^2 L^2 F' B U^2 L^2 U$
	Plummer's cross $R^2 L' D F^2 R' D' R' L U' D R D B^2 R' U D^2$
	Anaconda $L U B' U' R L' B R' F B' D R D' F'$
	Python $F^2 R' B' U R' L F' L F' B D' R B L^2$
	Black Mamba $R D L F' R L' D R' U D' B U' R' D'$
	Green Mamba $R D R F R' F' B D R' U' B' U D^2$
	Female Rattlesnake $U^2 D' L^2 D B U B' R' L^2 U^2 F U' F R$

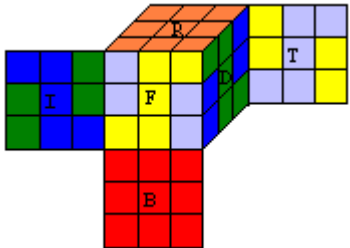
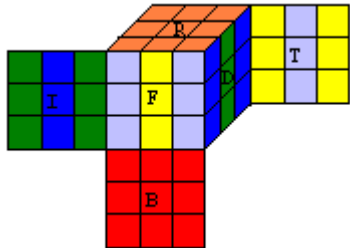
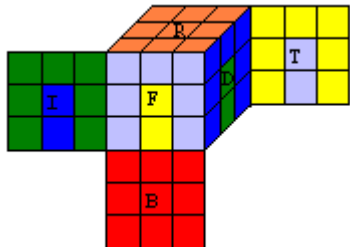
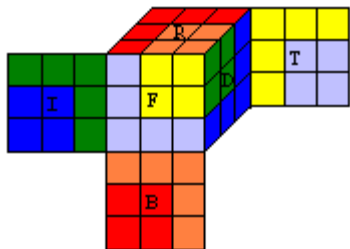
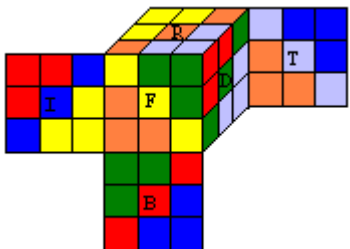
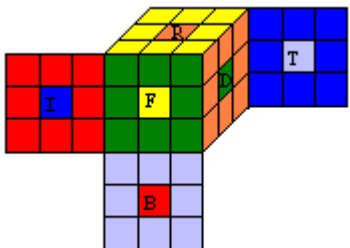
	Male Rattlesnake $R' F' U F' U^2 R L^2 B U' B' D' L^2 U^2 D$
	Female Boa $R U' R^2 U^2 F D^2 R^2 U' D' R D' F'$
	Male Boa $F D R' U D R^2 D^2 F' U^2 R^2 U R'$
	Four spot $F^2 B^2 U D' R^2 L^2 U D'$
	Six spot $U D' R L' F B' U D'$
	Orthogonal bars $F R' U L F' L' F U' R U L' U' L F'$
	Six T's $F^2 R^2 U^2 F' B D^2 L^2 F B$

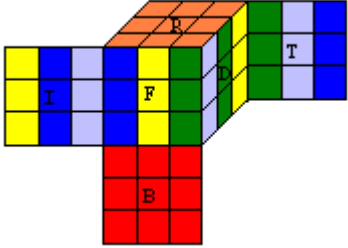
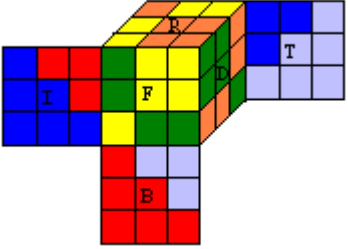
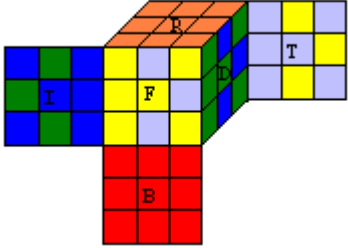
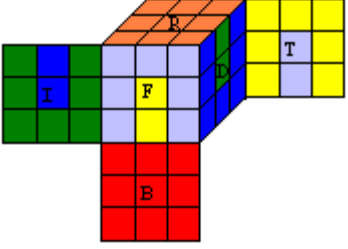
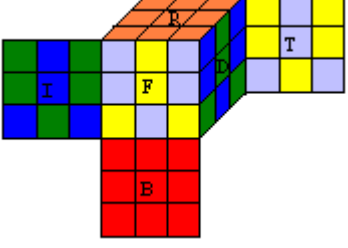
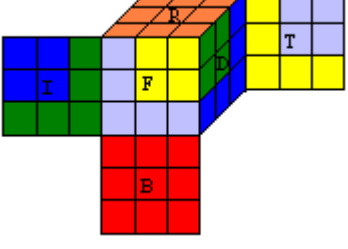
	Six-two-one $U B^2 D^2 L B' L' U' L' B D^2 B^2$
	Exchanged peaks $F U^2 L F L' B L U B' R' L' U R' D' F' B R^2$ $F^2 R^2 D R^2 U D F^2 D' R' D' F L^2 F' D R U'$
	Two twisted peaks $F B' U F U F U L B L^2 B' U F' L U L' B$ $F D^2 B R B' L' F D' L^2 F^2 R F' R' F^2 L' F'$
	Four twisted peaks $U' D B R' F R B' L' F' B L F R' B' R F' U' D$
	Exchanged chicken feet $F L' D' B' L F U F' D' F L^2 B' R' U L^2 D' F$
	Twisted chicken feet $F L' D F' U' B U F U' F R' F^2 L U' R' D^2$
	Exchanged rings $B' U' B' L' D B U D^2 B U L D' L' U' L^2 D$ $F U D' L' B^2 L U' D F U R^2 L^2 U' L^2 F^2$

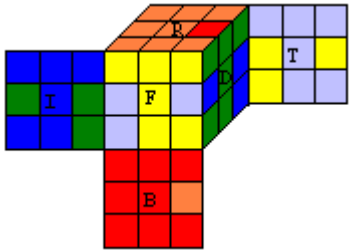
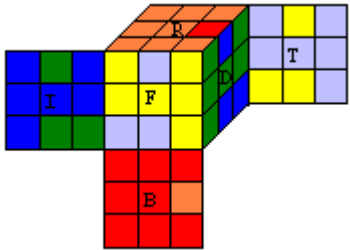
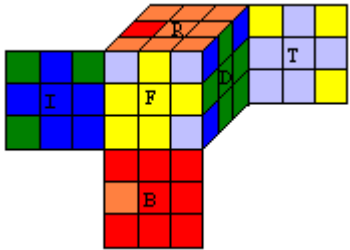
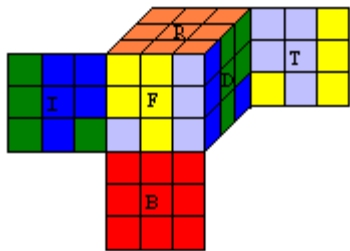
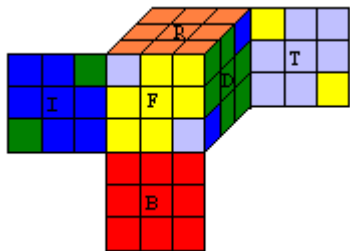
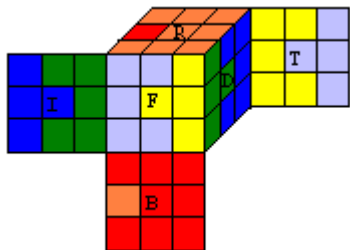
	Twisted rings $F D F' D^2 L' B' U L D R U L' F' U L U^2$
	Edge hexagon of order 2 $U B^2 U' F' U' D L' D^2 L U D' F D' L^2 B^2 D'$
	Edge hexagon of order 3 $D L' U R' B' R B U^2 D B D' B' L U D'$ $F L B U L F^2 B^2 R' F^2 B^2 U' B' L' F'$
	Tom Parks' pattern $L U F^2 R L' U^2 B' U D B^2 L F B' R' L F' R$
	Ron's cube in a cube $F D' F' R D F' R' D R D L' F L D R' F D'$ $L^2 D^2 L' D^2 B^2 L^2 B^2 L' D^2 L^2 B^2 L' B^2$
	Twisted duck feet $F R' B R U F' L' F' U^2 L' U' D^2 B D' F B' U^2$
	Exchanged duck feet $U F R^2 F' D' R U B^2 U^2 F' R^2 F D B^2 R B'$

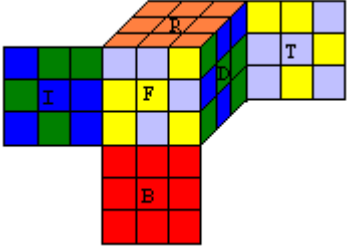
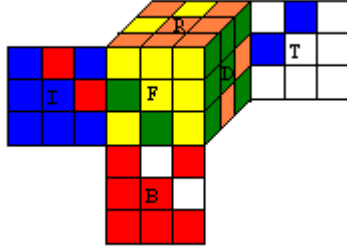
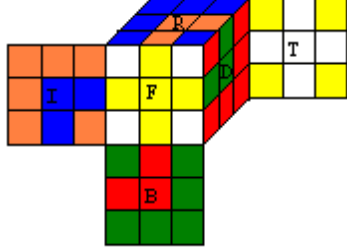
Jorge Jaramillo's pattern:

	Sin Centros D2 R2 B2 I2 F2 B2 R2 T2 B2 R2
	Letras H D2 I2 B2 D2 I2 B2 KR D2 I2 B2 D2 I2 B2
	Cruces (Haches) y luego R2 B
	Letras Y (Letras H) y luego B2
	Diagonales D I T F D I T F D I T F
	Zetas D2 R2 B2 I2 F2 B2 R2 T2 B2 R2 D I T F D I T F D I T F R2 B2

	Eses (Diagonales) y luego R2 B2
	Columnas (Sin Centros) luego el 2
	Enes D I R2 D- I- T F R2 T- F-
	Cubo dentro de Cubo T I- B2 I B F- B2 F B- T- F- D R2 D- R- T R2 T- R F
	Tres colores por cara T2 I F R- I- R- T- F I B- R T- R- F- I- R
	Todos los centros D I- F T- R B- D I-

	Tres Columnas: F T I F2 R B- I2 T- R- B- F2 D- R- B I- D B
	Bombón: T F- D- B T2 B- D F2 T2 R I- T2 I R- T F-
	Letras K: D2 B2 D2 B2 D2 B2 T2 B2 T2 B2 T2 B2 D I R2 D- I- F- T- R2 F T
	NUNU: D2 B2 R2 I2 F2 B2 R2 T2 R I2 D2 R2 I2 D2 R
	Cohete: D2 R2 B2 I2 F2 B2 R2 T2 B2 R2 D2 I2 B2 KD D2 I2
	Cubo en Cubo 2: I D F B- T I D F B- T I D F B- T D2 B2 R2 I2 F2 R2 B2 T2 R2 B2

	Letras J: T F I R- D T F I R- D T F I R- D- R2 B2 I2 F2 R2 B2 T2 B R D2 R- B T2 I B2 R2 D2 R2 B2 I
	Números 4: D2 I2 R2 D2 I2 F2 T2 R2 F2 T- F I R- D T F I R- D T F I R- D R B- D2 R- B T2 B2 R2 I B2 R2 D2 B2 R2 I B2 R2
	Moño: D2 I2 R2 D2 I2 F2 T2 R2 F2 T- F I R- D T F I R- D T F I R- D R B- D2 R- B T2 B2 R2 I B2 R2 D2 B2 R2 I
	Letras P: T F I R- D T F I R- D T F I R- D- R2 B2 I2 F2 R2 B2 T2 B2 R2 D2 I2 R2 D2 I2 F2 T2 R2 F2 T2 B2 D I B2 D- I- T- F- B2 F T
	Números 8: T F I R- D T F I R- D T F I R- D- R2 B2 I2 F2 R2 B2 T- F D B- I T F D B- I T F D B- I R2 B2
	Cruz en esquina: D- I- B2 I D- R2 B2 I2 F2 R2 B2 T2 R2 B2 KD2

	Tortuga: R D2 F2 B2 F2 B- F2 D2 B2 D2 R F2 R2 D I- B2 D I R2
	Corbatines: R T2 I T F- I2 D- T- F B R2 I- T2 R-
	El Ocho (8) B- I2 T2 F2 R2 I- R2 F D2 R- I2 F2 R2 D R2 T- R2

第二十二节 Bob Burton's Finger Tricks

Stabilizing Grips

Left standard grip: I grip the cube so that my left thumb is covering the lower 2x2 block on the front. My left ring and left middle fingers cover the corresponding 2x2 block directly behind the cube.

Right standard grip: This is the same grip as above, except using the right hand to cover the 2x2 blocks (on the right side of the cube).

Left top grip: This is the same grip as Left standard, except covering the corresponding 2x2 blocks on the top and bottom faces, instead. My left thumb covers the bottom left back 2x2 block and my left middle and ring fingers cover the top left back 2x2 block.

Right top grip: This is the same grip as above, except using the right hand to cover the 2x2 blocks (on the right side of the cube).

Slice grip: I grip the cube so that my left thumb is on the FL and FDL stickers and my left index and middle fingers are on the BL and BLD stickers, respectively.

(R U) triggers

Trigger	Stabilization	Grip	Execution
(R U)	Left standard	My right thumb is placed on the DR sticker. I place my right index, middle, and ring fingers on the URF, UR, and UBR stickers, respectively.	To perform the R, I rotate my right wrist away from me. After the R-move is complete, I pull the U-move with my right index finger.
(R U R')	Left standard	I grip the cube exactly the same way as I would for performing (R U).	I perform this trigger the same way as I perform (R U), but as the U-move comes close to completion, I begin to rotate my right wrist towards me again to finish the trigger.
(R U R2')	Left standard	I grip the cube exactly the same way as I would for performing (R U).	I perform this trigger the same way as I perform (R U R'), but I pull my wrist a little further towards me to complete the trigger.
(R U2 R')	Left standard	I grip the cube exactly the same way as I	I perform this trigger the same way as I perform (R U R'), but after the U-move,

		would for performing (R U).	I immediately follow through with my middle finger to perform the second U-move. I finish the trigger in the same manner.
(R U R' U)	Left standard	I grip the cube exactly the same way as I would for performing (R U).	I perform this trigger the same way as I perform (R U R'), but as the R'-move comes close to completion, I perform another U-turn by pulling with ring finger.
(R U R' U')	Left standard	I grip the cube exactly the same way as I would for performing (R U).	I perform this trigger the same way as I perform (R U R'), but after the R'-move is complete, I pull the U' with my left index finger.
(R U R2' U')	Left standard	I grip the cube exactly the same way as I would for performing (R U).	I perform this trigger the same way as I perform (R U R2'), but after the R2'-move is complete, I pull the U' with my left index finger.

(R U') triggers

Trigger	Stabilization	Grip	Execution
(R U')	Left standard	My right thumb is placed on the DFR sticker. I place my right middle and ring fingers on the UR and UBR stickers, respectively. My right index finger floats freely in the air.	To perform the R, I rotate my right wrist away from me. As the R-move comes close to completion, I begin to push with my right thumb to perform the U'.
(R U2')	Left standard	I grip the cube exactly the same way as I would for performing (R U').	I perform this trigger the same way as I perform (R U'), but I push further with my right thumb until the U-face has rotated twice, instead of just once.
(R U' R')	Left standard	I grip the cube exactly the same way as I would for performing (R U').	I perform this trigger the same way as I perform (R U'), but at the completion of the U'-move, I bring my wrist back towards me. You cannot cut the corner for final move in the trigger.

(R U' R)	Left standard	I grip the cube exactly the same way as I would for performing (R U').	I perform this trigger the same way as I perform (R U'), but as the U'-move comes close to completion, I continue to rotate my wrist away from me to finish the trigger.
(R U2' R')	Left standard	I grip the cube exactly the same way as I would for performing (R U').	I perform this trigger the same way as I perform (R U' R'), but instead of pushing my thumb for only U', I continue to push until I have performed U2', and then I use the side of my thumb to perform the R' as I bring my wrist back towards me.
(R U' R' U')	Left standard	I grip the cube exactly the same way as I would for performing (R U').	I perform this trigger the same way as I perform (R U' R'), but I follow the last R'-move with a pull of my left index finger to perform U'.

(R' U) triggers

Trigger	Stabilization	Grip	Execution
(R' U)	Left standard	My right thumb is placed on the UR sticker. I place my right index, middle, and ring fingers on the DRB, DR, and DFR stickers, respectively.	To perform the R', I rotate my right wrist towards me. As the R'-move comes close to completion, I begin to pull with my right index finger to perform the U.
(R' U2)	Left standard	I grip the cube exactly the same way as I would for performing (R' U)	I perform this trigger the same way as I perform (R' U), but after I pull the first U with my right index finger, I follow through and pull U again with my right middle finger.
(R' U R)	Left standard	I grip the cube exactly the same way as I would for performing (R' U)	I perform this trigger the same way as I perform (R' U), but as soon as I finish

			pulling the U, I rotate my wrist away from me to finish the trigger.
(R' U2 R)	Left standard	I grip the cube exactly the same way as I would for performing (R' U).	I perform this trigger the same way as I perform (R' U2), but after the U2 is complete, I rotate my wrist away from me to complete the trigger.
(R' U R')	Left standard	I grip the cube exactly the same way as I would for performing (R' U).	I perform this trigger the same way as I perform (R' U), but as the U-move comes close to completion, I continue to rotate my wrist towards me to finish the trigger.
(R' U2 R')	Left standard	I grip the cube exactly the same way as I would for performing (R' U).	I perform this trigger the same way as I perform (R' U2), but as the U2-move comes close to completion, I continue to rotate my wrist towards me to finish the trigger.

(R' U') triggers

Trigger	Stabilization	Grip	Execution
(R' U')	Left standard	My right thumb is placed on the UBR sticker. I place my right index, middle, and ring fingers on the DRB, DR, and DFR stickers, respectively.	To perform the R', I rotate my right wrist towards me. After the R'-move is complete, I push the U'-move with my right thumb.
(R' U' R)	Left standard	I grip the cube exactly the same way as I would for performing (R' U').	I perform this trigger the same way as I perform (R' U'), but as the U'-move comes close to completion, I begin to rotate my wrist back away from me to finish the trigger.
(R' U' R U)	Left standard	I place my right thumb on the FR sticker. I place my right index, middle, and ring fingers on the BRU, BR, and BDR stickers, respectively.	To perform the R', I rotate my wrist towards me. After the R'-move, I perform U' by pulling with my left index finger. As the U' comes close

			to completion, I rotate my wrist back away from me to perform the R. Finally, after the R, I pull with my right index finger to perform U.
(R' U' R U')	Left standard	I place my right thumb on the FR sticker. I place my right index finger on the UBR sticker and I place my right middle, ring, and pinky fingers on the BRU, BR, and BDR stickers.	To perform the R', I rotate my wrist towards me. After the R'-move, I perform U' by pushing with my left index finger. As the U' comes close to completion, I rotate my wrist back away from me to perform the R. Finally, after the R, I push with my right thumb to perform the last U'.

(R2 U') and (R2' U) Triggers

Trigger	Stabilization	Grip	Execution
(R2 U')	Left standard	I place my right thumb on the BDR sticker. I place my right index, middle, and ring fingers on the FRD, FR, and FUR stickers, respectively.	To perform the R2, I rotate my right wrist away from me. : As the 2nd R-move comes close to completion, I begin to push with my right thumb to perform the U'.
(R2' U)	Left standard	I place my right thumb on the BR sticker. My right index finger hooks around the FRD corner, so it is ready to push the U face when necessary.	To perform the R2', I rotate my right wrist towards me. As the 2nd R'-move comes close to completion, I begin to pull with my right index finger to perform the U.
(R2' U2)	Left standard	I grip the cube exactly the same way as I would for performing (R2' U).	I perform this trigger the same way as I perform (R2' U), but after I pull the first U with my right index finger, I follow through and pull U again with my right middle finger.
(R2' U R')	Left standard	I grip the cube exactly the same way as I would for performing (R2' U).	I perform this trigger the same way as I perform (R2' U), but as the U-move comes close to completion, I

			continue to rotate my wrist towards me to finish the trigger.
--	--	--	---

R,F Triggers

Trigger	Stabilization	Grip	Execution
(R' F R F')			

(M' U), (M' U'), and (M U)Triggers

Trigger	Stabilization	Grip	Execution
(M' U)	Slice	I place my thumb directly opposite my left, covering the FR and FRD stickers. I place my right index, middle, and ring fingers on BRU, BR, and BDR stickers, respectively.	To perform the M', I pull the BD sticker with my left middle finger. After the M'-move is completed, I pull the U with my right index finger.
(M' U2)	Slice	I grip the cube exactly the same way as I would for performing (M' U).	I perform this trigger the same way as I perform (M' U), but after I pull the first U with my right index finger, I follow through and pull U again with my right middle finger.
(M2' U)	Slice	I grip the cube exactly the same way as I would for performing (M' U).	I perform this trigger the same way as I perform (M' U), but after I pull the first M' with my left middle finger, I follow through and pull M' again with my left ring finger.
(M2' U2)	Slice	I grip the cube exactly the same way as I would for performing (M' U).	I perform this trigger the same way as I perform (M2' U), but after I pull the first U with my right index finger, I follow through and pull U again with my right middle finger.
(M' U M)	Slice	I grip the cube exactly the same way as I would for performing (M' U).	I perform this trigger the same way as I perform (M' U), but after the U-move is completed, I push the FD

			sticker with my left thumb to complete the trigger.
(M' U')	Slice	I grip the cube exactly the same way as I would for performing (M' U).	To perform the M', I pull the BD sticker with my left middle finger. After the M'-move is completed, I pull the U' with my left index finger.
(M' U' M)	Slice	I grip the cube exactly the same way as I would for performing (M' U).	I perform this trigger the same way as I perform (M' U'), but after the U'-move is completed, I push the FD sticker with my left thumb to complete the trigger.
(M U)	Slice	I grip the cube exactly the same was as I would for performing (M' U).	To perform the M, I push the FD sticker with my left thumb. After the M-move is completed, I pull the U with my right index finger to complete the trigger.

第二十三节 Rubik's Revenge

The **Rubik's Revenge** (also known as the **Master Cube**) is the 4×4×4 version of Rubik's Cube. Invented by Péter Sebestény, the Rubik's Revenge was nearly called the Sebestény Cube until a somewhat last-minute decision changed the puzzle's name to attract fans of the original Rubik's Cube.

Permutations

There are 8 corner cubelets, 24 edge cubelets and 24 centre cubelets.

8 Corners	$8! \times 3^7$	
24 Centres	$24! / (4!)^6$	每个面的四个中心块是等同的
24 Edges	$24!$	楞块不能翻转
Divided by	24	整体没有固定朝向，用第一个角块定朝向，有 24 种等价情形

总的置换状态数： $8! \times 3^7 \times (24!)^2 / [(4!)^2 \times 24] = 7.40 \times 10^{45}$

大纲思想是降群。降阶法即是降为三阶魔方群。还有其他的降群方法。这里先提供降为三阶魔方群这一熟悉子群的方法。

记 $GR = \langle L, R, U, D, F, B \rangle$

$G_0 = \langle L, R, U, D, F, B, M_0, M_1, E_0, E_1, S_0, S_1 \rangle$

$G_1 = GR \times \langle M_0, M_1, E_0, E_1, S_0, S_1 \rangle$

$G_2 = GR \times \langle M_0, M_1, E_0, E_1, S_0, S_1 \rangle$

$G_3 = GR \times \langle M_0, M_1, E_0, E_1, S_0, S_1 \rangle = GR$

$G_4 = \langle I \rangle$

这里把中间两层靠近 L 面得叫做 M_0 ，靠近 R 面得叫做 M_1 ， M_0 ， M_1 的转动方向跟三阶里的 M 方向一致。其他类推。

$G_0 \rightarrow G_1 \rightarrow G_2$ 降群的过程就是将三对对色（BG, WY, RO）的中心块分开，分别放到 LR, UD, FB 面的中心。

$G_2 \rightarrow G_3$ 降为三阶魔方群，这个过程就是将中心块的对色再分开，分别放到相应面心上，同时，将相邻楞块正确配对。

$G_3 \rightarrow G_4$ 就是三阶群的还原过程。下面分步介绍每一步需要考虑的问题，和解决的关键技巧。

One: $G_0 \rightarrow G_1$

注意观察八个角块的配色方案，它决定了中心块的摆放位置。跟三阶的一样即可，我定位为
上黄下白，左蓝右绿，前红后橙。

这一步做的是分离出 WY 色，放到 UD 面上。

技巧是，将 WY 色中心块两个两个的配对，方便再放回 UD 面上。比如先在 U/D 面上各放一对
WY 色块，剩余四个 WY 色块放到侧面配成两对，一个转动同时归位。

Two: G1-->G2

把 BG 色中心块放到 LR 面上，RO 色中心块放到 FB 面上。

这一步完成之后相对色的中心块，就不会在相邻面上，只有这样后续步骤才能在 G2 群内完
成。

关键技巧：利用 E0，E1 转动分离 BG 和 RO 色。显然的是，一次 E0，E1 转动对调了 4 个
LR 面上的中心块到 FB 面上，所以先调整 FB 面，使得 FB 面上 8 个中心块 4 个 RO 色，4 个
是 BG 色，其中，4 个 BG 色两个在 F 面另两个在 B 面上。LR 面上的情形跟 FB 面一致。调
调整好 RO 色块 BG 色块相对位置，一个 E0 或 E1 转动即可分离 FB，LR 面心。

Three: G2 -->G3

这一步把对色也分离，即做好六个面的中心。同时，将相邻楞块正确配对。配中心和配楞块
理应同时实现的，但太过复杂，为了避开庞大的计算量，我分两步完成。先配中心，再配对
楞块。

G3 群里的 (M0M1) 表示 M0，M1 两层绑定在一起同时转动，相当于三阶里的 M。因为 G1，
G2 群都能分离相邻位置的两个楞块，而 G3 群保持相邻楞块始终相邻，所以，楞块配对只
能在这一步里面实现。

1.配中心

技巧：比如配 FB 面的中心，先调整使得 F 面上有两个红色面，两个橙色面，同色的处于邻
位而非对角位，B 面中心也如此，再用 M02，或 M12 一次转动即配好 FB 面的中心。

2.楞块配对

技巧：将需要配对的楞块调到 M0，M1 层上来，做 (M0L) 或 (M1R) 转动配对楞块，然后
GR 群转动把配好的对调到边层，再作前面 M0，M1 转动的逆，复原 M0，M1 层。我这里 (M0L)
表示把 M0 层和 L 层绑定在一起做 M0 方向的转动。

通常，配对把两对错误棱放到 Ma，Mb 中间层，把另外一对错误的放在 L 或 R 层用于置换
配好的那对。所以最后剩下两对错误棱，就没有可供利用的指环棱。解决办法是，当剩下四
对错误棱时，要对配对精确设计：调整剩下四对成为 ab，ba，cd，dc 这样的情形，将 ab，
ba 对放到中间层的对角位置，Ma2 转动，配成 aa，bb 正确形式，用 cd，dc 把它们置换走，
然后操作 Ma2 恢复中心块的同时，将 cd，dc 错开，配成 cc，dd 正确形式。

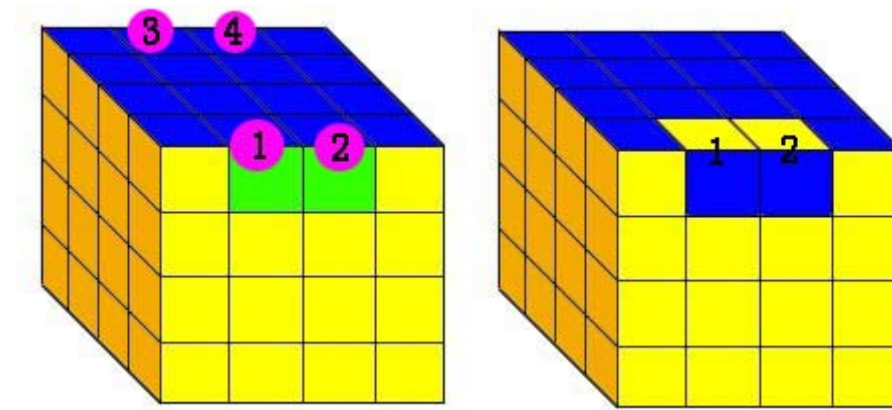
这一步还有另一种解决方法，总可以配 10 对最后只剩下两对棱，这时用一个三轮换，就可
把这四块配好对。

Four: G3 -->G4

最复杂的工作到这里基本做完，从十几种三阶解法随便选一个复原即可。

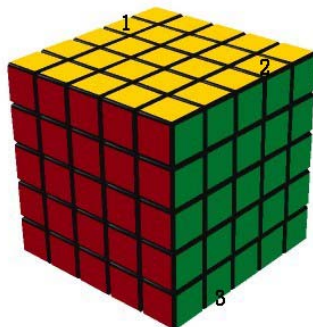
Five: Switch Edges Pairs

最后，会出现和三阶不一样的情形。



Case1.

第一种情况，看似要将 12 对，和 34 对，整体对换。实际上是 14 对换，23 对换。两个对换可以用三轮换实现，(LUSc,RUSc,DRSa)也即图示 (123):U'Ma'U-R2-U'MaU-R2



将的一种情型调整成这个三轮换形式，做两次三轮换，这四个棱即正确配对。用数学来讲： $(421)(132) = (14)(23)$ 。

当然，用对易子实现这个对换思路清晰但步骤太多，更简单的公式是：**Mb2 U2 Mb2 U2- Ea2 Mb2 Ea2**

Case2.

第二种情形看起来像同时翻转 12 两个楞块，四阶楞块不可能原地翻转，实际上是要对换 12，对换过程中朝向自动正确。这是个奇置换不可能用对易子实现。需要利用公式：

Mb2 B2 U2 Ma U2 Mb U2 Mb' U2 F2 Mb' F2 Ma' B2 Mb2

Pretty Patterns

4 parallel strips: Mb2L2-Sa2B2-Mb2L2

Cube in a Cube in a cube:

L'D2RD2R'FRU'DLFB'U'BD'L2U(Mm)'(Ee)'(Ff)(Rr)'(Ff)(Uu)(Rr)2(Ff)2(Dd)'(Bb)'(Uu)(Rr)'

记号：**Mm** 指中间 **MaMb** 的组合层，**Ff** 指 **F** 与其相邻的 **Sa** 的组合层，以此类推。

第二十四节 Proffesor's Cube

The **Professor's Cube** is a mechanical [puzzle](#), a $5 \times 5 \times 5$ version of the [Rubik's Cube](#).

There are 98 pieces on the exterior of the cube: 8 corner cubelets, 36 edge cubelets (two types), and 54 center cubelets (48 movable of two types, 6 fixed).

There are 98 pieces on the exterior of the cube: 8 corner cubelets, 36 edge cubelets (two types), and 54 center cubelets (48 movable of two types, 6 fixed).

Permutation

corner cubelets: $8! \cdot 3^7$

Center cubelets: $(24!/(4!^6))^2$

Central Edges cubelets: $12!/2 \times 2^{11}$

(an odd permutation of the corner cubelets implies an odd permutation of the central edge cubelets, and vice versa, thus the division by 2).

Outer Edge Cubelets: $24!$

(cannot be flipped)

The totle number of permutation is

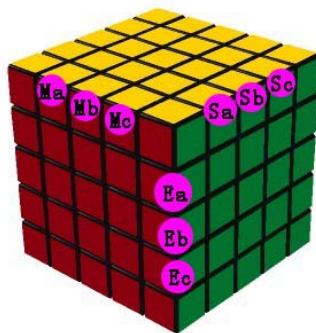
$$8! \times 3^7 \times 12! \times 2^{10} (24!)^3 / (4!)^{12} = 2.83 \times 10^{74}$$

Solution

同四阶：还原中心—楞块配对—三阶。

一些技术细节：

我的中间层标记方法如下图示，其转动方向与三阶的 M，E，S 方向相同。



middle edges and corners can be treated as equivalent to a $3 \times 3 \times 3$ cube.

9 个中心块，一个正中心块，位置固定，剩下八个分为两组，每一组内四个块完全等同，不可区分。

同色的两个边棱（outer edge），外观完全相同，但内部结构并不等同。它两个对换位置后朝向

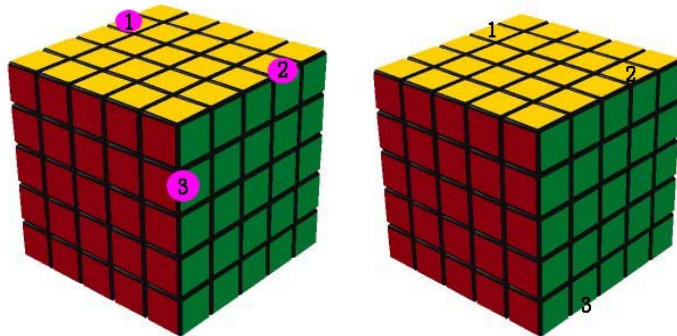
都变为错误，而不可能原地翻转，故而是可区分的。

还原到最后会遇到需要对换两个边楞块（Outer Edge）的情形，这是个奇置换。任何对易子都是偶置换，故而不可用楞块对易子实现。又因为颜色相同的两个棱（Outer Edge）不等同，无法用三个楞的三轮换。

一些三轮换：

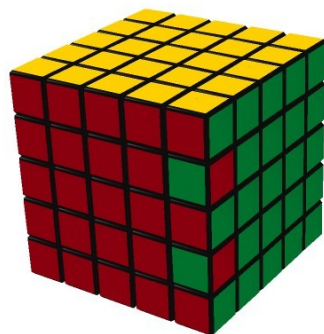
(LUSc,RUSc,FREa):U'Ma'U-R-U'MaU-R'

即下图中的轮换（1-->2-->3-->1）



将上面对易子中的 R 改成 R2，就变成轮换(LUSc,RUSc,DRSa):U'Ma'U-R2-U'MaU-R2。

上面所说有三个楞块不配对，如下图所示：



这时需要对换(FREa,FREc)两个楞块(注意不是分别翻转这两个楞块)，需要用到公式：

(FREa,FREc): Ec'R2-Ea'F2-EaF2-EcR2-EcR2-Ec'R2-F2Ec2F2

四阶的情形完全一样，也用这个公式即可把两个相邻楞块对换。

第二十五节 V-Cube 7/6

The **V-Cube 7** is the 7×7×7 version of [Rubik's Cube](#). It was invented by Panagiotis Verdes and is produced by his company, Verdes Innovations SA.

Patterns

六面六个心形图案:

B2D'U'LR-(EbEc)'-MeMa-(EdEe)'-McEa'Ee-MdMb-(EaEbEcEd)-(MaMbMcMdMe)'-R'L'UDB2

第二十六节 Pocket Cube

The **Pocket Cube** (also known as the **Mini Cube**) is the 2×2×2 equivalent of a Rubik's Cube. The cube consists of 8 corner pieces, and no other types of cubies.

Any permutation of the 8 corner cubies is possible (8! positions), and 7 of the cubies can be independently rotated (3⁷positions). There is nothing identifying the orientation of the cube in space, reducing the positions by a factor of 24. The number of possible positions of the cube is:

$8! \times 3^7 / 24 = 3674160$

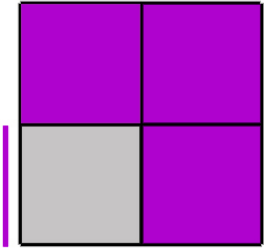
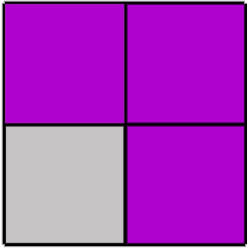
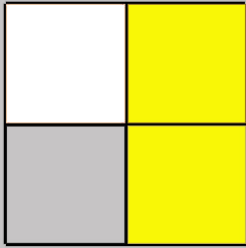
二阶有层先（FL，COLL），最流行的两种解法是 Guimond method 和 Ortega methods。

martijn's Doc for Guimond method and Ortega methods.

Guimond method

Step 1: Find 3 stickers of opposite color on 1 face.

The purple stickers can be various color, white/yellow, orange/red, or green/blue.

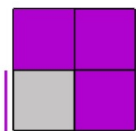
		
Table 1	Table 2	Example

This is already done 90% of the time. Else you have to set it up with one or 2 moves

When you have found one of the 2 patterns do a 'x2' move (turn the cube around) to get one of the patterns in table 1 or 2.

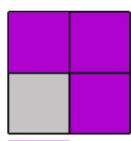
If you have table 1 pattern look there, and the same for table 2.

Step 2.1: Table 1



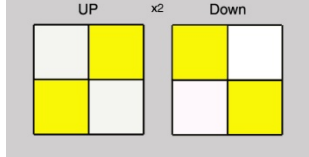
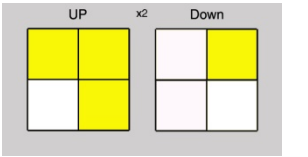
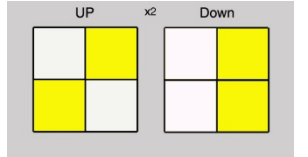
(y2) RU'R'	(y')R'F'R	F'L2F	L2'UL'
(x2) RU'BL'	R2F2UF	F'RU'F2U'	RF2U'R2FU'

Step 2.1: Table 2



(y)R'UR	LUL'	(y2)RU2R'	(y)L2D'R
RU'BL'	F2R2U'R'	R'FU'F2U	xUR'U'RU2R

Step 2.2:

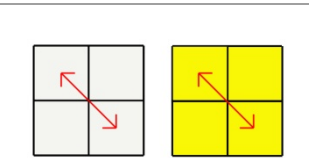
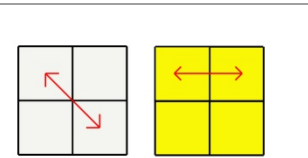
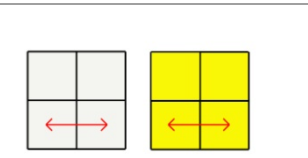
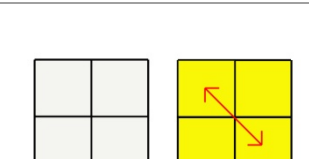
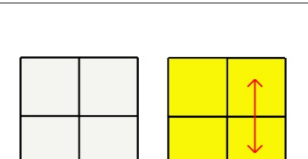
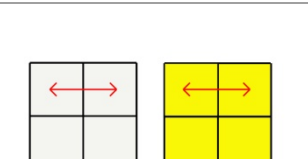
		
R2 U2 F2	R2 U' R2	R2 U' R2 U' R2

Step 3 – PLL (Up + Down)

You may have to turn the cube around to get the right pattern.

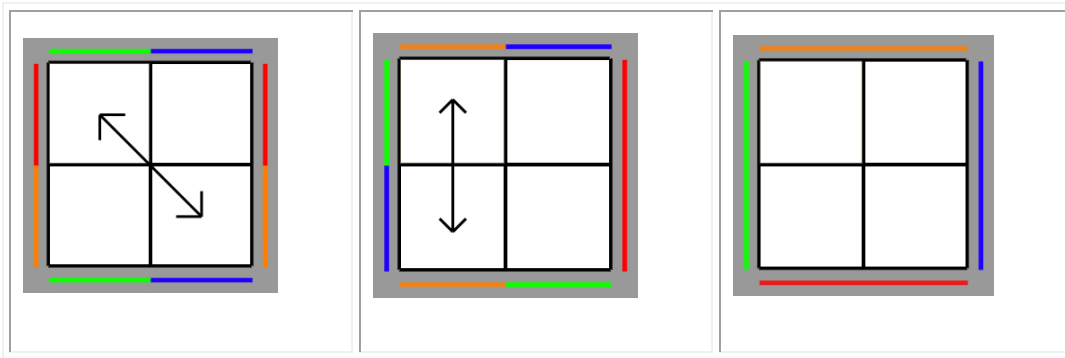
You may have to do a U, U' or U2 turn to get the pattern.

And most of the time you will have to do a U, U' or U2 turn at the end to solve the cube.

		
R2 F2 R2	(RU'R) F2 (R'UR')	R2 U F2 (U2 R2 U R2)
		
(RU'R'U') F2 (U'RUR') D R2	x U2(R'U'R)U2 x (R'UR'U'R2)	(R2 U R2) y' (U2 R2 U R2)

Ortega method

Step 1: Make the white face complete. The Permutation of the corners doesn't matter.



These are the 3 possible situations for the D face. (Color and place can be different)

Solve the Corners the same as on the 3x3. (LBL method)

To get the corners from the U-layer to the D-layer:

White Right - $R U R'$

White Left - $L' U' L$

White Up - $(R U^2 R') U' (R U R')$

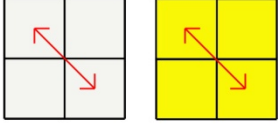
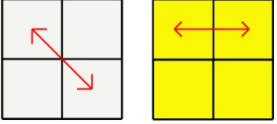
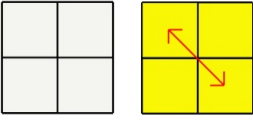
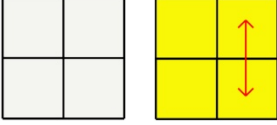
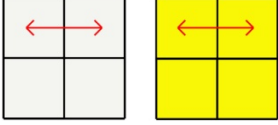
Step 2 - OLL

$(R'U'R)U'(R'U^2R)$	$(R'U^2R)U(R'UR)$	$F'(RUR'U') R'FR$	$(RUR'U') (R' FRF')$
$F (RUR'U') F'$	$F (URU'R') F'$	$F (RUR'U' RUR'U') F'$	$R^2 U^2 R U^2 R^2$

Step 3 - PLL (Up + Down)

You may have to turn the cube around to get the right pattern.

You may have to do a U, U' or U2 turn to get the pattern.
 And most of the time you will have to do a U, U' or U2 turn at the end to solve the cube.

		
R2 F2 R2	(RU'R) F2 (R'UR')	R2 U F2 (U2 R2 U R2)
		
(RU'R'U') F2 (U'RUR') D R2	x U2(R'U'R)U2 x (R'UR'U'R2)	(R2 U R2) y' (U2 R2 U R2)

第二十七节 粽子 Master Pyramorphix

四色粽子

Number of combinations

Corners: $8! \cdot 3^4$

Edges: $12! \cdot 2^{11} / (3!)^4$

Centers: 4^6

Even permutation factor: 2

总的状态数: $8! \cdot 3^4 \times 12! \cdot 2^{10} \times 4^6 / (6)^4 = 5.06 \times 10^{18}$

正四面体的外形，包裹着三阶的结构。

区别有三点：中心块中心块有方向，角块分为大角块小角块两组，与小角块相邻的三个楞块等同（小角块没有方向）。第二点减弱了三阶的对称性，第三点增加了三阶的对称性。

玩法跟三阶一样。可以先不管中心块朝向按三阶方法复原。之后再调整中心块朝向。

旋转中心块朝向：

1. U 面逆时针 F 面顺时针旋转 90 度: $(U' D' L' R' F' B) U' (F B' L' R' U D') F$
2. U 面顺时针 F 面逆时针旋转 90 度: $U' D' L' R' F' B) U (F B' L' R' U D') F'$
3. U 面 F 面同时旋转 180 度: $(U' D' L' R' F' B) U_2 (F B' L' R' U D') F_2$
4. U 面中心块旋转 180 度: $(L R U_2 L' R' U) \times 2$

直接按照三阶解法复原，识别判断很麻烦。Jaap 提供一个方法，先后将角块，楞块，中心块的方向调整好，将粽子的外形复原，然后再还原对应块的位置。这个方法前半部分可以用来解单色粽子。

单色粽子

单色粽子十二个楞块完全相同，可互相替换，但每个棱块都有方向，共有有 2^{11} 种朝向。四个大角块（或四个小角块）也完全相同，可互相替换，也没有方向，可随意扭转。角块有 $8! / (4!4!)$ 种组合方式。中心块可以转 90 度，有两个方向，6 个中心块共有 2^6 种朝向。同时，魔方块整体不能奇置换，除以 2。所以，单色粽子的组合方式总共有 4, 587, 520 种组合方式。跟二阶魔方的组合方式数目一个量级。远少于四色的 5.06×10^{18} 种组合方式，也远少于普通三阶的 4×10^{19} 种组合方式。

根据 Jaap 的方法，单色粽子还原可以分简单的三步：

1.还原四个大角块，这一部很简单，不需要任何公式。

2.翻转楞块。对于粽子来说，它等价于几个楞块的置换。把需要对调的两个楞块 180 度转动到同一个面 U 面上，只有两种情形：

翻转相邻两楞块 UR-UB: RDB^2RD , 或者用 $(R'URU')^2$ 。起一个公式步骤少, 后一公式好记易操作。以 $(R'URU')^2$ 为例, 它实际上是 UR, UB, RB 三个楞块的轮换 ($UR \rightarrow UB \rightarrow RB \rightarrow UR$), 同时翻转了与这三个楞块相邻的四个角块, 对于单色粽子来说, 总体效果是翻转 UR, UB 这两个楞块。

翻转相对的两楞块, UF-UB: $R'FRUR'U'F'R'$

3. 旋转中心块, 只有 90 度旋转一种情形, 只需要一个公式。

旋转 U 面中心块: $R'URU'F'U'F$

Master Pyramorphix Solution

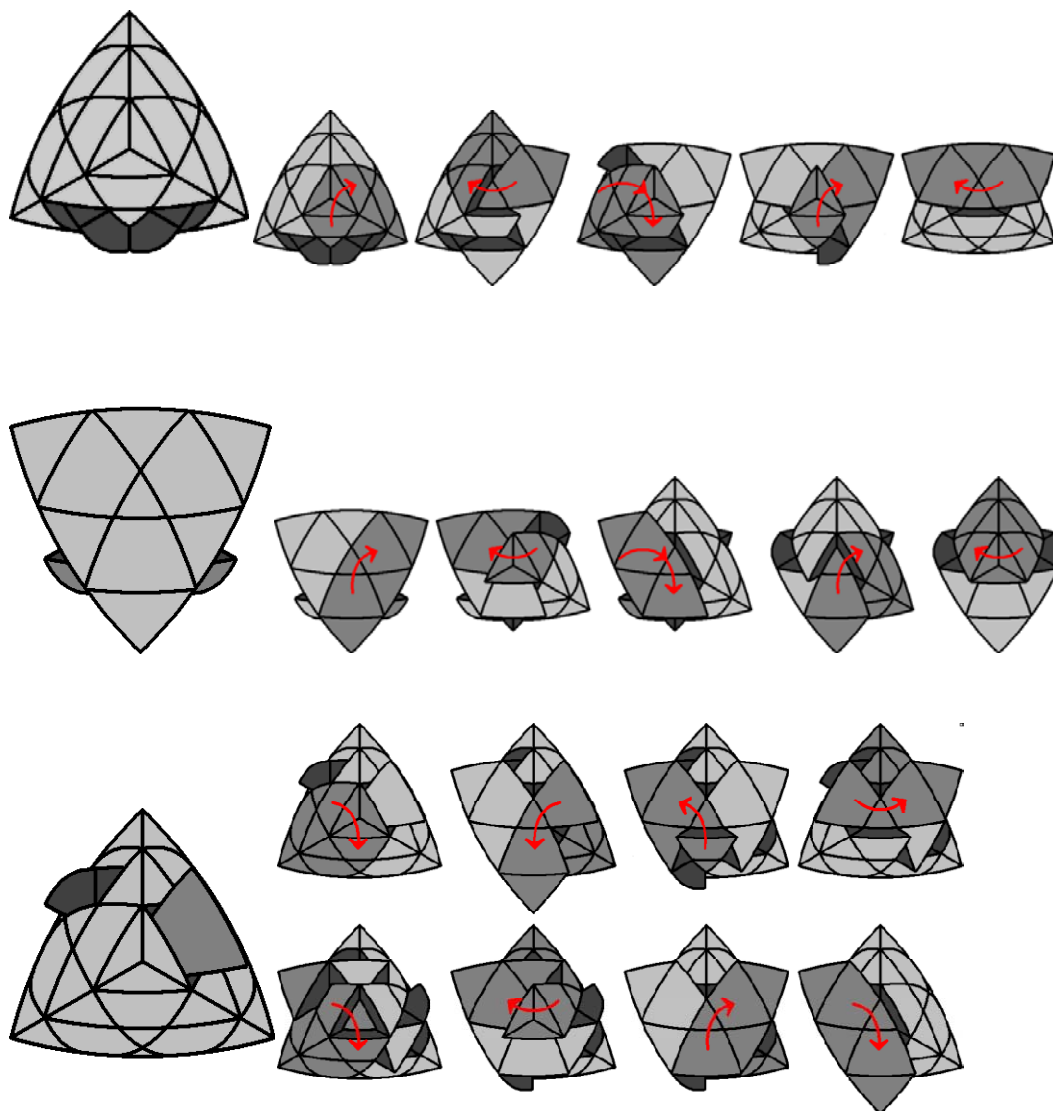
By Jaap Scherphuis

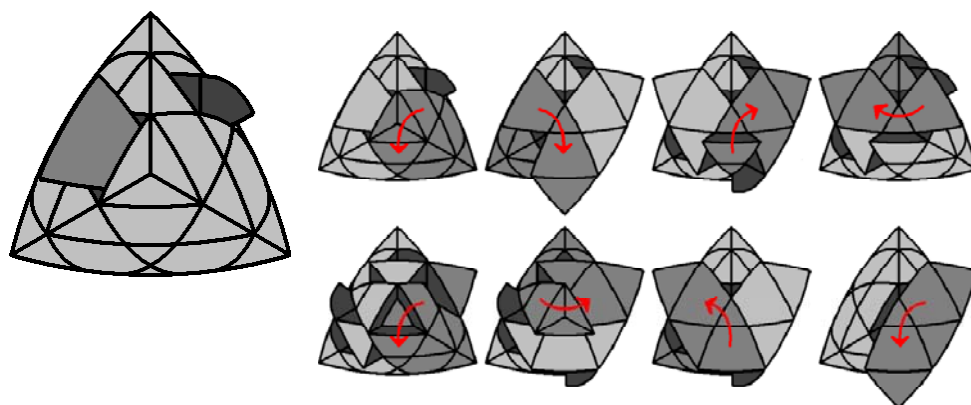
1. Corners in shape

The four big corners can be placed so that they form a pyramid shape in just a few moves.

2. Orient the twelve wedge pieces

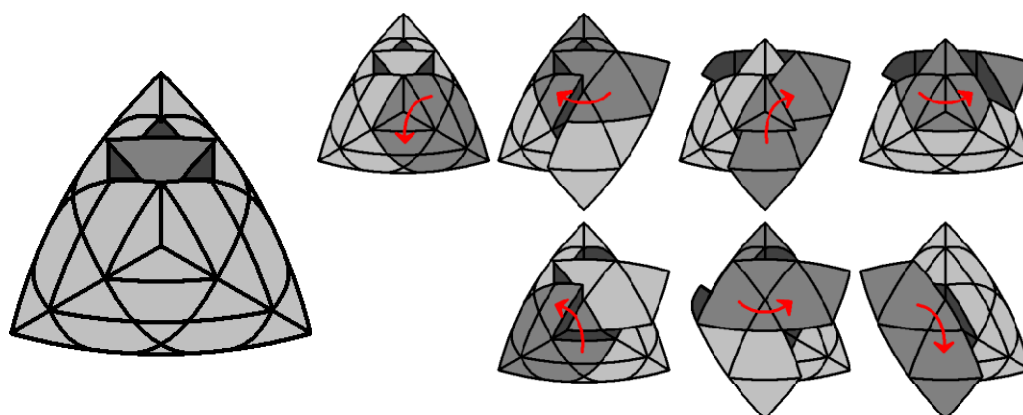
The move sequences below will flip two adjacent wedge pieces. If the two wedge pieces you want to flip are not close to each other, then you can use half-turns to bring them together.





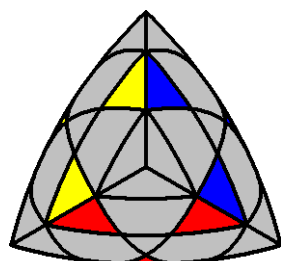
3. Orient the six ridge pieces

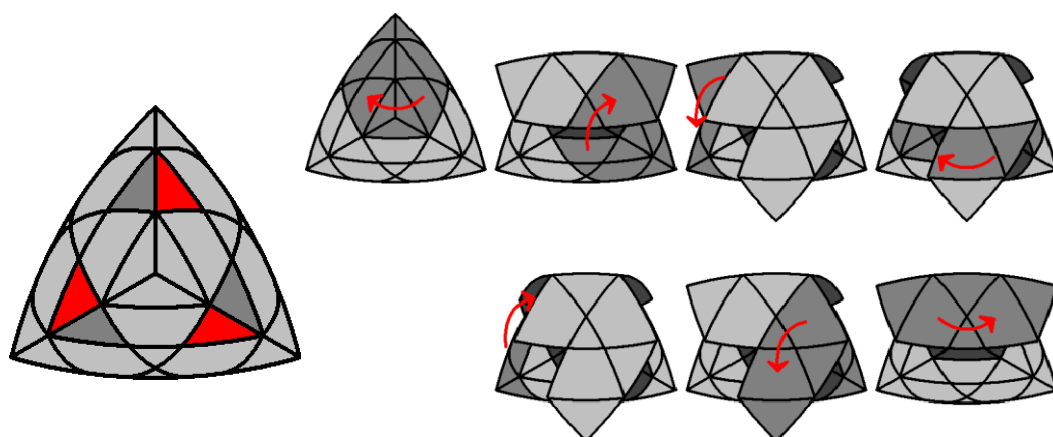
With the move sequences below you can twist a single ridge piece. After orienting these the puzzle should be in the correct shape.



4. Solve the six ridge pieces

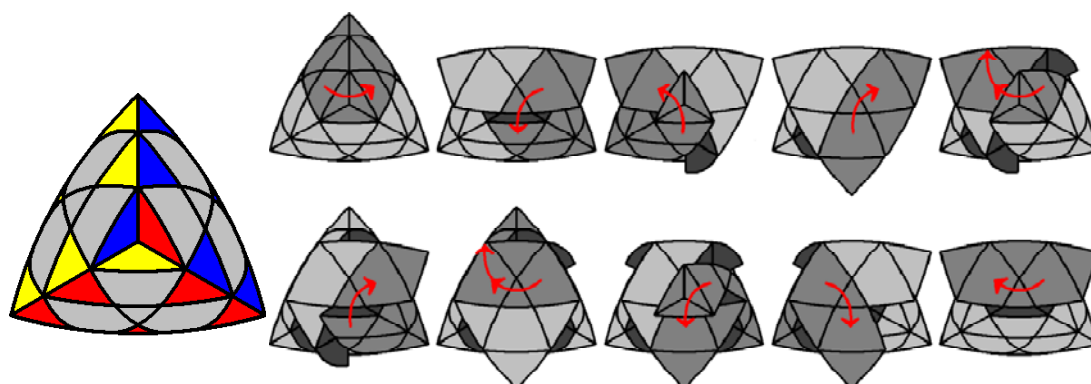
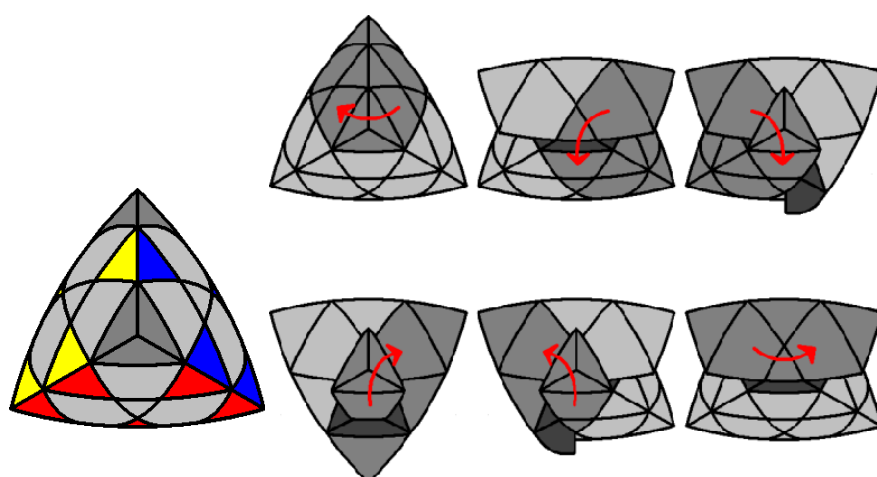
Any ridge piece can be given a half-turn without affecting the other ridge pieces. Look at the three ridge pieces that share a face. If they have a colour in common, then you can turn them to show that colour in the shared face. If they do not have a colour in common, do the sequence below and try again. The remaining ridge pieces can then be easily solved.





5. Solve the four corner pieces

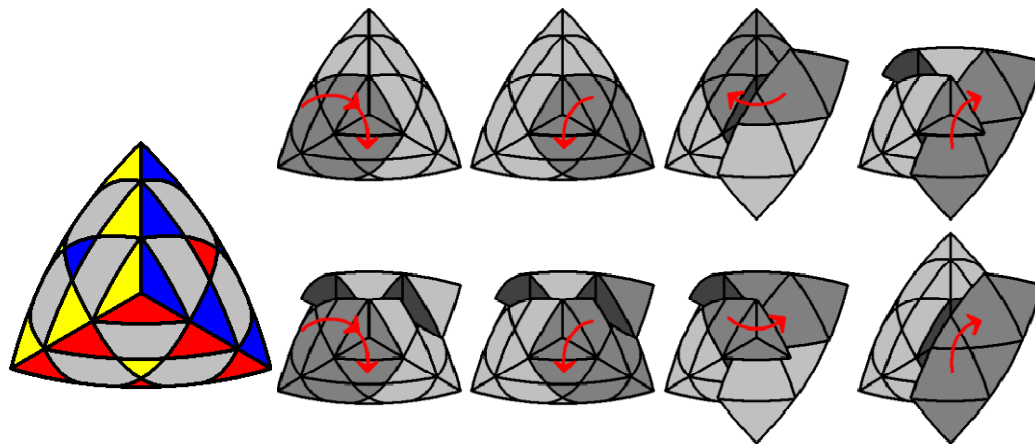
If a corner is out of place, then you can use the first sequence below to put it in its correct place. It swaps two corners without affecting the other two. Once a corner is in the correct place, you can use the second sequence to orient it properly.



6. Solve the four triangle pieces

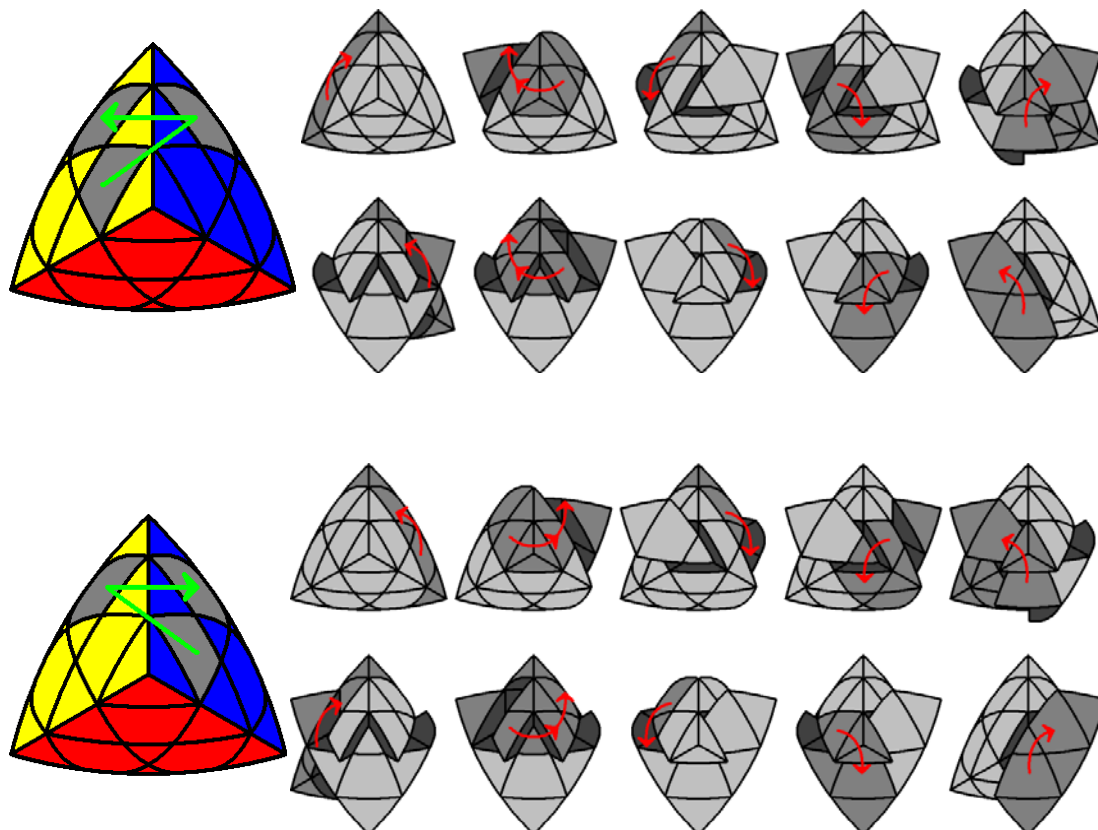
The four triangle pieces can be put in place using the sequence below which swaps around

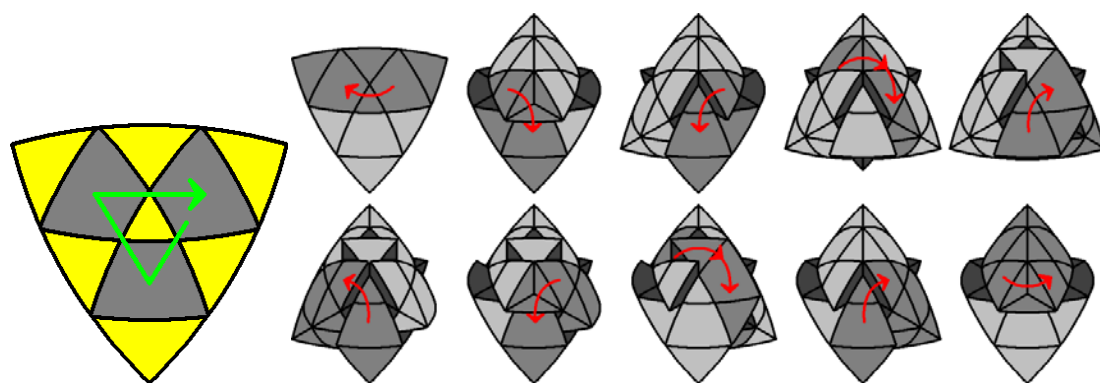
three of the triangles. If all four triangles are wrongly placed, you will have to use this sequence once to solve one triangle, and a second time for the remaining three triangles.



7. Solve the twelve wedge pieces

You can move wedge pieces from one face to another using the first two move sequences below. The third move sequence moves around the wedge pieces inside a face. To swap two wedge pieces in adjacent faces, you can use the third sequence if necessary to move the two pieces towards where the faces meet, and then the first or second sequence once or twice to swap them around.





第二十八节 五魔方 Megaminx

The Megaminx, or Magic Dodecahedron, was invented by several people simultaneously and produced by several different manufacturers with slightly different designs.

The Megaminx is made in the shape of a dodecahedron, and has 12 face center pieces, 20 corner pieces, and 30 edge pieces. There are two main versions of the Megaminx: the 6-color version in which opposite faces have the same color, and the 12-color version in which all faces have a different color.

There are $20!/2$ ways to arrange the corners and 3^{19} ways to orient them, since the orientation of the last corner depends on that of the preceding ones. There are $30!/2$ ways to arrange the edges and 2^{29} ways to flip them.

So, the total number of combinations is:

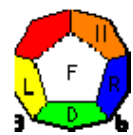
$$20! \times 3^{19} \times 30! \times 2^{29} / 4 = 1.01 \times 10^{68}$$

五魔方可以很简单的类似 F2L 逐块逐面还原，最后只剩下一个面，分步还原楞块角块位置和方向。

最后一面可用类似 8355 处理顶层方法，留下侧面的一个角块楞块对和顶层一块还原。先还原 6 个棱，再还原最后 6 个角。可先用三循环复原 6 个角块位置，再翻转角块方向。

下面是 Jaap 的五魔方还原方法。

Notation:



Let U, R, F be three adjacent faces (Up, Right and Front), in other words, U, R, F are faces arranged clockwise around a vertex. Let D be the other face adjacent to both F and R. Clockwise $1/5$ twists of a face are denoted by the appropriate letter, anti-clockwise $1/5$ twists by U', R', F' or D'. In the solution below, the D face is usually still unsolved. We can denote the positions of the edge pieces by using two letters, for example UF is the edge that lies in both the U and F faces. Similarly, corners are denoted by three letters as they lie in three faces.

Solution:

If you can solve the top and middle layer of the Rubik's cube, then you can use the same techniques on the megaminx to solve everything but the final layer.

Phase 1: Solve all but final face

- a. Solve the edge pieces of one face. This is easy to do and need not be explained in detail. Let that face be the U face.
 - b. Solve the corners of the U face as follows:
 1. Find a corner that belongs on the U face.
 2. Hold the puzzle so that the corner belongs at URF, and that face D is unsolved.
 3. Use any turns that do not disturb already solved pieces to place the corner at RDF.
 4. Use one of the following sequences to place the corner:
 RDF->URF: R'D'R
 FRD->URF: FDF'
 DFR->URF: R'DR FD'D'F'

These sequences only disturb the FR edge piece and the D layer, but nothing else except for the corner. The sequences differ only in the orientation of the corner. If you use the wrong sequence for your situation, the URF corner can be twisted in place by R'DR FDF'.

 5. Repeat 1-4 till done. If necessary you can use to use this method to displace a corner that you need somewhere else.
 - c. Place all yet unsolved edges adjacent to the solved corners. This is similar to placing the corners, except that you use the following sequences to place them:
 DR->FR: DFD'F' D' R'DR
 FD->FR: D'R'DR D FD'F'
- These sequences only disturb the D layer, but nothing else except for the edge. If an edge is in position but flipped, then displace it by placing any other edge piece there with one of the sequences, and then insert the edge piece in the correct orientation.
- d. Repeat steps b-c to solve each adjacent face in turn, until only one unsolved face is left.

In the phases below, we need to have names for all the faces adjacent to the D face in order to describe all the possible pieces on the D face. Lets call these faces in clockwise order: L, F, R, b, B.

Phase 2: Orient the edges of the final face.

The edges are flipped in pairs. It is impossible to have only one edge upside down.

- a. Hold the puzzle so that the unsolved face is D, one edge to be flipped is at DR, and another is either at DL or DF.
- b. Do one of the following sequences to flips the edges:
 To flip DF, DR, do R D F D' F' R'
 To flip DL, DR, do R F D F' D' R'
- c. Repeat the above if necessary.

Phase 3: Position the edges of the final face.

- a. Rotate D to place as many edges as possible in their correct places. If, like Alexander's Star, your megaminx has only 6 colours, you might find that the edges are in an odd permutation. In this case you will have to swap any two identical edges elsewhere on the puzzle and try solving it again.
- b. Hold the puzzle so that you can use one of the following sequences to solve it:
 1. DR->DF->DL->DR: R D R' D R D'D' R'

2. DR->DL->DF->DR: R DD R' D' R D' R'
3. DR->DF->DB->DR: R D R' DD R DD R'
4. DR->DB->DF->DR: R D'D' R' D'D' R D' R'
5. DR-DB, DF-DL: R D R' D R D' R' DD R DD R'

Phase 4: Position the final corners.

- a. Hold the puzzle so that you can use one of the following sequences to solve it:
 1. DFR->DLF->DBL->DFR: RUR' D RU'R' D RUR' D'D'RU'R'
 2. DFR->DBL->DLF->DFR: RUR' DD RU'R' D' RUR' D' RU'R'
 3. DFR->DLF->DbB->DFR: RUR' D RU'R' DD RUR' DD RU'R'
 4. DFR->DbB->DLF->DFR: RUR' D'D'RU'R' D'D'RUR' D' RU'R'
 5. DFR-DbB, DLF-DBL: RUR' D RU'R' D RUR' D' RU'R' DD RUR' DD RU'R'

Note the similarity between these sequences and those of phase 3.

Phase 5: Orient the final corners.

- a. Hold the puzzle so that the DFR corner needs to be twisted anti-clockwise. If there is no such corner, then put a different twisted corner there.
- b. To turn DFR anti-clockwise, and another clockwise, do one of these:
 1. DFR->RDF, DLF->LFD: RU'R'F'U'F D F'UFRUR' D'
 2. DFR->RDF, DBL->BLD: RU'R'F'U'F DD F'UFRUR' D'D'
 3. DFR->RDF, DbB->bBD: RU'R'F'U'F D'D' F'UFRUR' DD
 4. DFR->RDF, DFb->FbD: RU'R'F'U'F D' F'UFRUR' D
- c. Repeat the above until all corners are correct.

Gigaminx: 五阶五魔方



第二十九节 SQ1,SSQ1

The Square One, also known as Back to Square One and Cube 21, is a puzzle similar to the Rubik's Cube.

Number of positions

If different rotations of a given permutation are counted only once while reflections are counted individually, there are $170 \times 2 \times 8! \times 8! = 552,738,816,000$ positions.

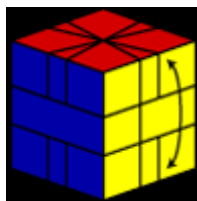
If both rotations and reflections are counted once only, the number of positions reduces to $15! / 3 = 435,891,456,000$.

一个上下层的转动对换了八块，控制起来太复杂。总结发现，只有上下层保持为方形时，每个面上有三对正交的切线，具有最大的转动自由度。切换块的时候，上下层要错开 30 度，这样是为了始终保持正方形。

观察可以看到，上层八个块可以分为四对，每个角块和它相邻的楞块作为一对，作为一个整体，上下层转动只能转 90 度角。中间层只是个障碍层，可以当做没有。这么一来，SQ1 就成了个二阶魔方。我虽然没玩过二阶的，但无疑二阶是可解得。这样一来，就有了还原它的思路，第一步是将角块和楞块正确配好对，然后按照二阶的玩法就可以还原。

配对还是有些麻烦的。哪两块应该配成一对，中间层的颜色判断，还是挺绕眼的。还有个问题是最后有两对没法在只转 90 度的情况下配对。暂且不管它，整体作为二阶的还原，最后结果就只需要交换一对相邻角块。这种情况需要找公式。

Notation(Lars Vandenberg):



Twisting the right hand side 180°

(a,b) where **a** denotes the multiple of 30° you have to turn the top layer and **b** denotes the multiple of 30° you have to turn the bottom layer.

/ means twisting the right hand side 180°.

Section 1:Lars Vandenberg's Solution to SQ1

This method has been used to set most Square-1 world records.

Step 1: Make both layers square

Step 2: Bring the corners in their correct layer

Step 3: Bring the edges in their correct layer

Step 4: Permute the corners within their layer

Step 5: Permute the edges within their layer

Step 6: Correct middle layer and swap layers if necessary

Section 2:Jason Baum's Solution to SQ1

I recently came up with my own method to solve the Square-1. I'm sure others have used something similar to this before, but I haven't seen anything like it online anywhere. Currently the most popular method to solve Square-1 is Lars Vandenberg's method. It involves orienting corners, orienting edges, permuting corners, and permuting edges. It is no doubt a great method, but the problem with it, in my opinion, is that the final step (permuting edges) is very difficult. There are 99 cases, and many of them are hard to recognize. When I decided I wanted to get better at Square-1, I briefly looked at this method and quickly decided I didn't want to devote the amount of time it takes to learn 99 algs for this puzzle. So I started thinking and pretty quickly came up with an alternative method that is more layer by layer style. The great thing about this method is a large portion of it is completely intuitive, and it also cuts the number of algs in half (only 50 required algs for the entire method) Additionally, the steps are very easy to recognize, and it has a nearly identical move count to Lars Vandenberg's method. I am very optimistic about the potential of this method. So optimistic, in fact, that I think a sub 20 average is possible with it.

1) Make both layers square(Intuitive)

2) Solve the bottom layer corners(Intuitive)

-2 optional cases

-You should also make sure the middle layer is fixed during this step

3) Solve the bottom layer edges

a) Solve two adjacent edges

-Intuitive

b) Solve the other two edges

-7 cases

c) Special Cases (optional)

-8 cases

-This step is a bit outdated, as some of the cases contain an odd number of twists. I need to re-generate those cases so that they have an even number of twists so that the middle layer stays in tact.

4) Permutation of the Last Layer

-43 cases (21 without parity, 22 with parity)

Section 3: Shellie's Solution to SQ1

How to Solve the Square-1

Everyone and their mom knows about the Rubik's cube, and a considerable subset of everyone and their mom knows how to solve a Rubik's cube. But what about the Rubik's cube's anemic red-headed cousin, the Square-1?

I would like to start by pointing out that there are other tutorials on the web that are much more comprehensive and helpful than my own (might I suggest Jaap Scherphuis's puzzle site or Lars Vandenbergh's CubeZone?). So why does this tutorial exist? In short, I got a Square-1 one day and discovered that I'm completely useless at learning Square-1 algorithms. After a few attempts with online guides, I gave up and decided to figure out how to solve it on my own. This is the very elementary method I worked out, using algorithms I made up as I bumbled my way to a solution. The algorithms provided will help you solve your Square-1, but not particularly efficiently. It's called the SSS1M, short for Shellie's Sillie Square-1 Method. Yes, it is so silly it involves a misspelling of my own name as well as the word silly.

Initially I never meant to publish this method, but Lucas Garron convinced me it would be funny. As such, I slapped it together at work one day. There are no pictures. Sorry. Use your imagination or find a better tutorial.

(As a side note, despite the self-proclaimed silliness of this method, Dan Dzoan learned it and was faster than me within 24 hours. Stephanie Chow has also since become a remarkably successful student of this tutorial. Take that as you will.)

Part 0: Finding your way around the Square-1

The top and bottom layers of the Square-1 consist of corner (big) and edge (small) pieces and can be rotated freely. The middle layer is divided into two halves, and the puzzle can be rotated vertically if all the pieces on the top and bottom layers line up along the cut. Notice that the radial angle of a corner piece is twice the radial angle of an edge piece.

Square-1 algorithms look like pairs of numbers in parentheses separated by forward slashes. Each number denotes the multiple of 30 degrees to turn the top or bottom layer (first number for top layer, second number for bottom layer; positive number for clockwise, negative number for counterclockwise). An easier way to think of this is to remember that each big piece counts as 2 and each small piece counts as 1. Each / means to turn the right side vertically 180 degrees, swapping the upper and lower layer

pieces on the right half of the puzzle. When following the algorithms, hold the puzzle so the middle layer has a smaller section on the left side.

Part 1: Making both layers square

Your first objective is to make the top and bottom layer square. I find it is easier to do this by first grouping all eight edges together on the top layer. For the most part, this can be done intuitively, but if you get stuck, you should check out a more comprehensive guide. I recommend you practice doing this step intuitively before learning to solve the rest of the puzzle; it makes things a lot easier.

Once you have all the edges grouped on the top layer, you should have a "horseshoe" shape on top and a "star" shape on the bottom. Position the top layer so that the line of symmetry of the "horseshoe" lies along the center cut and turn.

Now you should have "paw" shapes on the top and bottom layer. Position them so that the line of symmetry lies along the center cut and both "paws" point in the same direction and turn.

Now you have two "barrel" shapes. Position one vertically and one horizontally and turn.

Now you have two "stingrays". Position them so that the center cut is along the long axis of the stingray (head to tail) and both "stingrays" are pointing in the same direction and turn to complete the square. Don't worry about the shape of the middle layer until the end.

In shorthand, start with the ends of the "horseshoe" pointing away from you and do $(-2,-4) / (-1,-2) / (-3,-3) /$

Part 2: Separate pieces into their proper layer

Start by putting the corners in their proper layer. This can be done intuitively. If you do a $(1,0)$ or $(0,-1)$ before you start and only turn the top and bottom layers in multiples of 90 degrees (multiples of 3 in Square-1 algorithm notation), you can preserve the square shape and treat the corners as a 2x2 cube.

Once the corners are in their proper layer, do the same for the edges. Position "wrong" edges at UB and DF and do $(0,-1) / (-3,0) / (4,1) / (-4,-1) / (3,0) /$

You can also swap two pairs of "wrong" edges at a time by positioning them at UF, UB, DF and DB and doing $(1,0) / (-1,-1) /$

Part 3: Solve corners

I use only one algorithm for this. It switches ULB/URB and DLB/DRB:

$(1,0) / (3,0) / (3,3) / (3,0) /$

Faster variant of this algorithm which I learned from Nick Young (it deviates briefly from the square shape, but don't worry):

$/ (3,0) / (-3,-3) / (0,3) /$

You may have to do the algorithm twice. Another one switches diagonal corners on both layers, but this case is rare:

$/ (-3,3) / (3,-3) /$

Part 4: Solve edges

I use two algorithms to solve edges. To swap UF/UB and DF/DB:

$(1,0) / (-1,-1) / (6,0) / (1,1) /$

The following does the equivalent of a Z-permutation on the Rubik's cube. It swaps UL/UF and UB/UR edges. Combine it with a $/$ setup move to swap bottom layer edges if necessary.

$(1,0) / (-1,-1) / (-3,0) / (1,1) / (3,0) / (-1,-1) /$

Note: Try $(0,-1) / (-3,0) / (1,1) / (2,-1) /$ to swap UL/UF and DF/DR. It's the same as the pseudo-Z-permutation above but with the additional setup move and with resulting extraneous moves truncated.

Part 5: Parity

Sometimes you will end up with parity, where you have exactly one pair of edges that need to be swapped. There are good ways to solve parity. This isn't one of them. Go back to Part 1 and work backward from the square to the "paw" shapes. Swap 3 corners on the top with 3 corners on the bottom with a $/$ turn, go back to your square shape, and start the solve over.

I told you this was an inefficient method.

Ideally, you should be able to spot whether or not you have parity before beginning step 4 (or step 3 if you're good at recognizing PLLs), so you don't have the entire solve wasted.

Part 6: Fixing the middle layer

Once your top and bottom layers are complete, fix the middle one if necessary. This part is intuitive, but I've included algorithms for those who can't figure it out on their own. To make the middle layer square if it isn't already, do $/ (6,0) / (6,0) /$. If the middle layer is upside down (which really means your top and bottom layer need to be swapped, do $/ (6,6) /$.

You're done!

Section 4: Jaap's Square-1 Solutions

Getting the cube shape

First you must get the puzzle into cube shape. This takes at most 7 twists (ignoring the centre layer shape), and the shortest method is simply using a table to look up the moves since there are only 90 different possible puzzle shapes. Such tables can be found at various websites, but they are not particularly enlightening. Instead I will present three methods here. The first is the most intuitive, and is the most common way of doing this phase. The second method is very different, but interesting as it shows an underlying mathematical constraint that I find quite surprising. The third is a little similar to the first though described in much more detail.

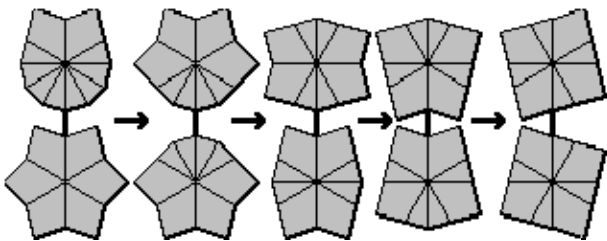
I have made a [diagram](#) showing the various shapes. At the top is the solved cube shape, and each successive row below it is one move further away. The blue shapes have 6 corners in one layer and two in the other, and these shapes are used in method 1 below. The red shapes have five and three corners in the layers. The yellow, green and purple ones have 4 corners in each layer, and the three colours signify the type of shape as used in method 2.

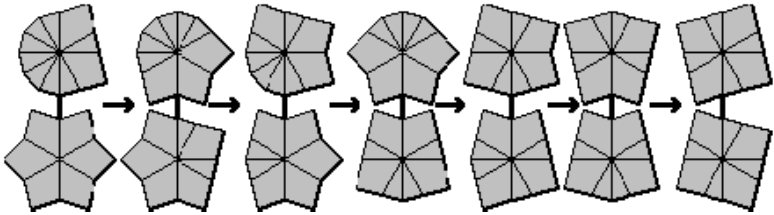
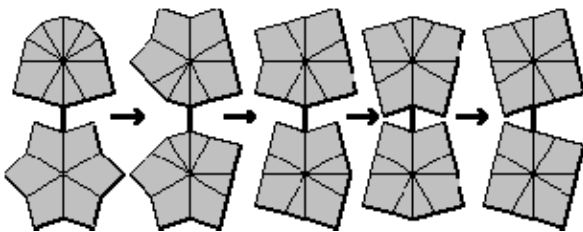
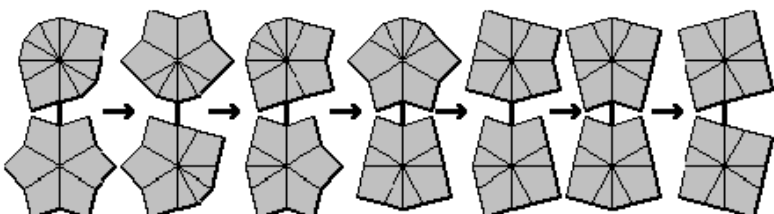
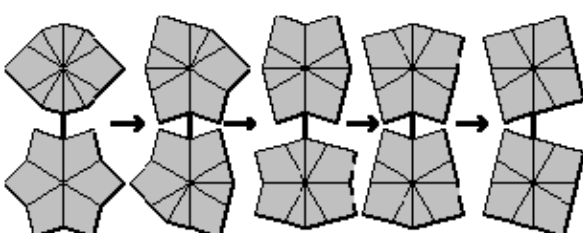
Method 1: Gathering corners

This method simply involves bringing six of the corners into one layer, which leaves the remaining pieces (2 corners and 8 edges) in the other layer. From such a position it is fairly easy to recover the cube shape.

I will not go into detail on bringing 6 corners together as it is quite easy. Once you have three corners together in half of one layer, then move these pieces around to the left half of the bottom layer. Now keep them there, so only use twists of the right hand side and turns of the top layer, and try to bring three more corners together in the top layer. Finally bring these two sets of three together in one layer.

Hold the puzzle with the six corners in the bottom layer. The top layer has two corners and six edges. There are only 5 possible shapes it can have, because there must be anywhere from 0 to 4 edge pieces between the two corner pieces. The diagrams below show how the cube shape can be recovered in each case.

0	$/(-2, -4) / (-1, -2) / (-3, -3) /$	
---	-------------------------------------	--

1	$/ (2, -2) / (-3, -4) / (4, -3) / (-5, -4) / (6, -3) /$	
2	$/ (-4, -2) / (-1, 4) / (-3, 0) /$	
3	$/ (-4, 0) / (5, 4) / (2, -3) / (-5, -4) / (6, -3) /$	
4	$/ (2, 2) / (0, -1) / (3, 3) /$	

The notation used here will be explained later when it will be more extensively used instead of diagrams.

Method 2:

This is a method found in the [rec.puzzles](#) archive.

There are 10 shapes with 4 edges (e), 4 corners (c) that a layer can have. Looking clockwise around the layer, they are:



cecece: Square 0



cceeccee: Barrel 2



ccceeee: Mushroom 0



cccccee: Scallop 2



ccceeece: Shield 0



cceceeece: Kite 2



cceeecece: Right Fist 1



cceceeece: Left Fist 3



ccceeece: Left Paw 1



ccceeece: Right Paw 3

If you are in a position with two of these shapes, look up the numbers associated with them in the list above and add them to get a number. Whatever move you do that keeps 4 corners in each layer, the number arrived at will not change (modulo 4). In particular if the number is not 0 (or 4) you cannot get it into a cube shape without first moving to a position that does not have 4 corners on each side. You then have to move back to a position with 4 corners in each layer and that does have a 0 or 4 sum.

If you do get a 0 (or 4) sum then the following list shows you how far away you are from the cube shape:

Square-Square		0	
Kite-Kite		1	
Barrel-Barrel	2	L.Fist-R.Fist	2
Shield-Shield	3	L.Paw-R.Paw	3
Scallop-Scallop	3	Barrel-Kite	3
Mushroom-Mushroom	3	Square-Shield	3
Barrel-Scallop	4	Square-Mushroom	3
		Kite-Scallop	3
		R.Fist-R.Paw	4
		L.Fist-L.Paw	4
		Mushroom-Shield	4

In the [diagram](#) of all the various shapes, the types of position can be seen. The yellow shapes have sum 0, the green have sum 2, and the purple have sum 1 or 3. The mirror image of a shape with sum 1 will have sum 3 and vice versa. Only one of each such pair is shown in the diagram. Note that shapes with only one non-symmetric layer will always have sum 1 or 3.

Method 3: Pairing the edges

I am indebted to Robert Richter for the following method of restoring the square 1 puzzle to cube form. I have worked out his method in great detail, because I find it an interesting approach. Once

you develop a feeling for it, it becomes easy and you will not need to remember any move sequences.

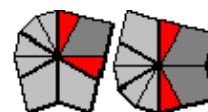
The main idea is to combine all the edge pieces into adjacent pairs of edges. Once this has been done, it is easy to bring all four pairs together and from there get the cube shape using the first move sequence in the first method above. The great advantage of having only pairs of edges is that they take exactly the same amount of room as corner pieces. Each layer therefore will consist of six parts of equal size (corners and/or edge pairs) which can be moved around without obstruction.

I'll explain below in more detail how to make edge pairs, but before I do it is important to be able to distinguish between edge pairs and single edges. This is usually really obvious, but sometimes when there are an odd number of edges together in a group it is not immediately clear which of them is the single edge and which ones form the pairs. A simple way to determine this is to remember that the single edge is always one of those at the outside of the group, never one in the middle. Often one of those outside edges cannot be moved because the seam that separates it from the others does not extend all the way through the layer. As it cannot be separated, it must form a pair with the one next to it, and the edge at the other end of the group is the single one. If both edges can be moved then either one of them (but not both) can be considered to be single.

Another way to determine the single edges is by seeing how close the layer is to being made of six equal sized parts. There are several shape types:

1. The layer is square, and obviously there are 4 single edges in this case.
2. The layer does divide into 6 equal sized parts. Here all edges (if any) already occur in pairs.
3. The layer has 4 of the 6 equal parts, but the rest consists of two edges with a corner between them. Those two edges are the single ones.
4. Half the layer consists of 3 equal parts, and the rest consists of two edges with two corners between them. Again those two edges are single.

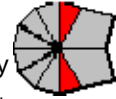
Sometimes a layer is ambiguous, because it belongs to both cases 3 and 4 above. In particular, the layer shape CCECEEEEE can be split as EE-EE-C-C-ECE, or as C-EE-EE-ECCE. We will use this ambiguity later on.



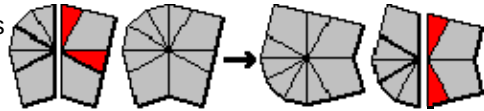
So, the first thing to do to move towards the cube shape is to pair up the edge pieces. Getting three pairs is fairly easy, in fact, in more than half the configurations there are already three pairs. In nearly all other cases there will be unpaired edges in both layers. It is then possible to bring an unpaired edge in the top layer to the front to the left of the seam, and an unpaired edge in the bottom layer to the back to the right of the seam so that a single twist brings the two edges together and does not split any already paired edges. This process can be repeated if necessary until you have at least three edge pairings. There is one special case you should be aware of, where all unpaired edges are in a single layer which only occurs when it is a square shape. In this case simply do any twist (that does not split any pairs in the other layer) to spread the unpaired edges to both layers and then continue the process outlined above to bring them together.

You will now have at least 3 pairs of edges. There are now three possibilities left:

1. You already have 4 pairs of edges. It is now a simple matter to bring these pairs together to form a large group of 8 adjacent edges.
2. You have two unpaired edges which are 150 degrees apart, i.e. they are separated by two corner pieces. Turn the layer with these two edges so that they both lie on the left hand side of the seam. I prefer to have them at the bottom left so I turn over the puzzle if they are on top. You next use only turns of the top layer and twists of the right hand side to move the three edge pairs together in a large group of 6 edges in the top layer. Often this is easy. The trickiest case is when the bottom right half consists of an edge pair between two corners. This pair cannot be immediately placed next to another edge pair, so you will have to twist it to the top and bring an edge pair to the bottom so that it lies next to the seam. Just be patient and make sure not to turn the bottom layer or to break up the edge pairs. Once you have the large group of 6 edges you can place them next to the two single edges to make a group of 8 edges together.

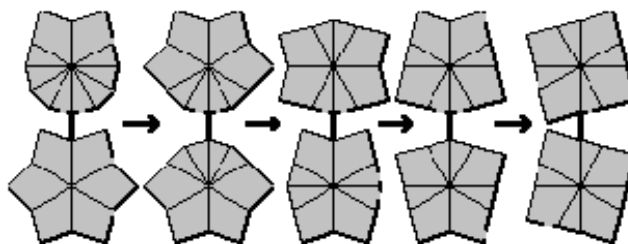


3. You have two single edges that are 90 degrees apart, i.e. they are separated by exactly one corner piece. Turn the layer with these two edges so that they both lie on the same side of the seam, and turn the puzzle so that they are in the left half of the bottom layer. Using only turns of the top layer and twists of the right half of the puzzle bring (at least) two of the edge pairs together in the top layer to make a group of 4 (or 6) adjacent edges. It is now always possible to bring the single edges and the group of 4 together in one layer to make the layer shape CCECEEEEE (or its mirror image CCEEEEECE), and this leaves (or moves) the remaining edge pair to the other layer. Because of the ambiguity of the edge pairings in this shape, you can consider the two edges on either side of the two adjacent corners to be the unpaired ones instead. This can therefore be solved as in case 2 above.



[Note: You could use the ambiguity of the CEEECCEEE layer shape instead but this it is slower to solve because its edge pairs are separated, and you need to put all three edge pairs together in case 2.]

At this point all eight edges are together in one layer. A single simple sequence of moves now recovers the cube shape. Split the group of edges in two, making both layers into the scallop shape (CCCCEEEE). Now split each set of edges in two again to get both layers into the barrel shape (CCEECCEE). Split one barrel between a pair of corners and the other between a pair of edges, to get both layers into the kite shape (CECECECE). If you split these between their edge pairs you form two squares.



The above should give you enough information to eventually get two squares. If the middle layer is not square, a single twist can make it so, though it is not yet necessary to do this at this point. The middle layer could well get disturbed in the later stages, but it can always be solved independently at the end.

Solving the pieces.

Notation:

From now on we need a notation for the moves. Hold the puzzle with the 'square 1' text on the left, so that the cut in the middle layer goes from the front left to the back right. I hold my left thumb on the small front-left face of the middle layer, and my left fingers on the back. I use my right hand to turn the top or bottom layers, as well as the right hand half of the puzzle.

Turns of the top and bottom layers are denoted by a pair of numbers (n,m) . They are the multiple of 30 degrees that the top/bottom layers are to turn respectively. A 3 means clockwise 90 degrees, 1 means 30 degrees (i.e. one edge along) and negative numbers mean anti-clockwise turns. A twist of the right hand side of the puzzle is denoted by a /. As the puzzle is in the cube shape now, the move sequences $(1,0)/(-1,0)$ or $(0,-1)/(0,-1)$ for example both leave the puzzle in cube shape if you ignore the middle layer.

As a shorthand, I will also use letters to denote the possible positions of the pieces in the cube. This notation is based on that of the Rubik's Cube. Let the faces be denoted by L, R, F, B, U and D (left, right, front, back, up, down). Positions of edge pieces can be denoted by two letters, for example the edge position on the left of the top layer is denoted UL. Similarly positions of corners are denoted by 3 letters, for example DFR is the front-right corner position of the bottom layer.

A short note about odd permutations:

Now that the puzzle is a cube shape, it is tempting to think that it can now be solved without changing shape. Many of the sequences used below indeed keep the top and bottom layers square. Each shape preserving move swaps two pairs of corners and two pairs of edges. This is an even number of swaps, so this is an even permutation. No combinations of even permutations can possibly be an odd permutation so any position that needs an odd number of swaps to solve it, cannot be solved without changing shape. In particular, to make a single swap of pieces you must change the shape of the puzzle. The quickest way to perform an odd permutation is to go to the scallop-scallop shape (3 twists), swap three corners from one layer with three from the other (1 twist), and return to the cube shape (3 twists) which takes 7 twists in total.

There are several methods for placing the pieces into their correct positions. I will show the most common ones here.

Solution 1:

This first method was worked out and given to me by Bobfish. The pieces are first put into their own layers, then the corners are placed, and finally the edges. Some of the sequences used in this

solution can be made slightly shorter. If you want, you can look them up in the computer generated tables that are listed in method 2 below. I have left this solution as is, because Bobfish found these sequences all by hand without computer. The parity fixer in particular is especially impressive - it is a much better clean odd edge swap than I ever found by hand myself.

Phase 1: Put pieces in their layers.

This phase puts all the white pieces in the top layer, and all the green pieces in the bottom layer.

You can always keep the top and bottom layers square throughout this phase.

- a. First put all the green corners in the bottom layer. This is quite easy if you first put two green ones in the left half of the bottom layer, then pair up the other two green corners and also twist them into the bottom layer.
- b. You can now make the middle layer square if necessary by doing the move sequence $(6,0)/(6,0)/(6,0)$. All move sequences later on will have an even number of twists, so will leave the middle layer square.
- c. Now bring the edges to their layer. There are 8 possible cases, which are listed below. In each case I will list the positions where there are edges in the wrong layer.
 1. All edges are in the correct layer: do nothing.
 2. UB, DL: do moves $(1,0)/(-3,0)/(-1,-1)/(3,0)/(1,-2)/(-3,0)/(-1,-1)/(4,1)/(-1,0)$
 3. UF, UB, DF, DB: do moves $(1,0)/(-1,-1)/(0,1)$.
 4. UB, UL, DR, DB: do moves $(1,0)/(-3,0)/(-1,-1)/(4,1)/(-1,0)$
 5. UB, UL, DF, DB: do moves $(1,0)/(3,0)/(-1,-1)/(-3,0)/(4,1)/(-3,0)/(-1,-1)/(4,1)/(-1,0)$
 6. UF, UB, DB, DL: do moves $(0,-1)/(-3,0)/(1,1)/(3,0)/(0,-3)/(-3,0)/(-1,-1)/(4,1)/(-1,0)$
 7. UF, UR, UB, DF, DB, DL: do moves $(1,0)/(-1,-1)/(-2,1)/(-3,0)/(-1,-1)/(3,0)/(1,-2)/(-3,0)/(-1,-1)/(4,1)/(-1,0)$, or apply sequence 3 and then sequence 2.
 8. All edges wrong: do moves $(1,0)/(-1,-1)/(-2,-2)/(-1,-1)/(0,1)$, or apply sequence 3 twice.

Phase 2: Put the corners correct.

This phase puts all the corners correct, while leaving the all edges in their own layers.

- a. This phase can be done using only two move sequences:
 1. Swap UFR-UBL, DFR-DBR: $(3,0)/(-3,0)/(3,0)/(3,0)/(6,0)/$
 2. Swap UFR-UBR, DFL-DBR: $(0,-3)/(0,3)/(0,-3)/(0,-3)/(0,6)/$
 If you want to keep the layers square, you may do $(1,0)$ before performing these, and $(-1,0)$ at the end.
- b. The second sequence is actually the same as the first performed upside down.
- c. You will probably have to apply these sequences more than once before the corners are solved.
- d. If you only need a single corner swap, then give that layer a quarter turn. Its corners now need a three-cycle, which can be done by twice swapping adjacent corners.

Phase 3: Put the edges correct.

- a. Most of this phase can be done using only two move sequences:
 1. Swap UF-UB, DF-DB: $(1,0)/(-1,-1)/(6,0)/(1,1)/(-1,0)$
 2. Swap UL-UF, DR-DF: $(1,0)/(-3,0)/(-1,-1)/(3,0)/(1,1)/(-3,0)/(-1,-1)/(4,1)/(-1,0)$
- b. You will probably have to apply these sequences more than once before the edges are mostly solved.
- c. These sequences all perform two swaps. It is possible that you need an odd number of swaps to solve the edges. To this end you will have to fix the parity by using the following move sequence:
 3. Swap UF-UB: $/(3,3)/(1,0)/(-2,-2)/(2,0)/(2,2)/(-1,0)/(-3,-3)/(-2,0)/(3,3)/(3,0)/(-1,-1)/(-3,0)/(1,1)/(-4,-3)$

Solution 2:

This method solves the corners completely, then the edges in their layers, and finally the edges completely.

Phase 1: The corner pieces.

In this phase, every twist will keep the layers square.

- a. Find the piece that should be at UFL. If it is in the bottom layer, then bring it to the top with a twist, (keeping the U,D layers square by turning either layer slightly before twisting). Then turn the top to place it correctly.
- b. Find the piece that should be at UBL. If it is incorrectly placed in the top layer, then bring it to the bottom with a twist. Then turn the bottom to place it at DFR. Now do $(4,0)/(-4,0)$
- c. Find the piece that should be at UBR. If it is incorrectly placed in the top layer, then bring it to the bottom with a twist. Then turn the bottom to place it at DFR. Now do the twist $(1,0)/(-1,0)$.
- d. Find the piece that should be at UFR. If it is not in place then turn the bottom to place it at DFR. Now do the sequence $(1,0)/(0,-3)/(0,3)/(-1,0)$.
- e. Rotate the bottom layer to place at least 2 corners are correct. If you need to swap two corners, then you can use the following two sequences:
 1. Swap DFR-DBR: $(1,0)/(0,-3)/(3,0)/(-3,0)/(-3,3)/(-3,0)/(2,0)$
 2. Swap DFL-DBR: $(1,0)/(3,3)/(0,3)/(3,3)/(0,3)/(3,3)/(-1,-3)$
 It is possible to perform the second (diagonal) swap by using the first (adjacent) swap twice.

Phase 2: Place the edge pieces in their correct layer.

Rotate the top/bottom layers to bring it to a position where one of the following sequences will place the edges in the correct layer. I have tried to include as many useful cases as possible in this table, but there is no need to memorise all of these. If you know only one of the sequences 8 to 13, you can use it repeatedly until the edges are in the correct layer, though using sequence 3 and 4 will speed things up.

1		Swap DF-UF, DR-UR, D B-UB, DL-UL	$(3,2)/(1,1)/(-3,3)/(-1,-1)/(0,-5)$	$[4^* 13^*]$
2		Swap DF-UF, DB-UB	$(0,5)/(1,1)/(-1,6)$	$[2^* 7^*]$
3		Swap DF-UB, DB-UF	$(0,-1)/(1,1)/(-1,0)$	$[2^* 6^*]$
4		Swap DR-UR, DB-UB	$(0,2)/(0,3)/(1,1)/(-1,-4)/(0,-2)$	$[4^* 11^*]$
5		Swap DR-UB, DB-UR	$(0,2)/(0,3)/(1,-5)/(-1,5)/(0,3)/(0,-2)$ $/(0,3)/(0,-4)/(1,1)/(-1,3)/(0,-3)/$	$[5^* 13^*]$ $[6 13^*]$
6		Swap DB-UB, DR-UF	$/(-3,0)/(0,5)/(6,1)/(0,3)/(-5,0)/(-1,6)$	$[6^* 14^*]$
7		Swap DB-UF, DR-UB	$(1,0)/(0,5)/(0,-3)/(-1,6)/(0,-5)/(0,3)/(0,6)$	$[6^* 14^*]$
8		Swap DR-UF, UR-UB	$(1,0)/(0,-3)/(-1,2)/(1,1)/(0,-3)/(0,3)/(-1,0)$	$[6^* 15^*]$
9		Swap DR-UR, UF-UB	$(1,0)/(0,-3)/(-1,-4)/(1,1)/(0,3)/(0,3)/(-1,0)$	$[6^* 15^*]$
10		Swap DR-UB, UF-UR	$(0,-1)/(0,3)/(0,-3)/(1,1)/(-1,2)/(0,-3)/(0,1)$	$[6^* 15^*]$
11		Cycle UF->UR->DR	$(-5,-3)/(5,0)/(0,3)/(-5,0)/(5,0)/(0,-3)/(0,3)$	$[6^* 14^*]$
12		Cycle UF->UB->DR	$(1,0)/(-1,0)/(0,3)/(-5,0)/(5,0)/(0,-3)/(1,0)/(5,0)$	$[7^* 15^*]$
13		Swap UF-DF	$(0,-3)/(0,3)/(1,0)/(2,-2)/(4,0)/(4,0)/(-4,4)/(0,1)/$ $(0,3)/(-3,-1)/(1,-2)/(-1,0)$	$[11^* 27^*]$

Phase 3: Place the edge pieces in each layer correctly.

Rotate the top/bottom layers to bring it to a position where one of the following sequences will place the edges correctly. There is no need to memorise all of the sequences, as only a few of the single layer swaps will be sufficient. Alternatively any odd permutation together with sequences a4 and a5 will do.

a. The following sequences perform swaps in both layers:

- 1  Swap UF-UB, UR-UL, DF-DB, D
R-DL $(1,0)/(2,-4)/(1,1)/(2,2)/(-5,1)/(5,0)$ [5*|15*]
- 2  Swap UF-UL, UR-UB, DF-DL, DR
-DB $(0,2)/(-3,0)/(1,1)/(3,-3)/(-1,-1)/(0,3)/(0,-2)$ [6*|16*]
- 3  Swap UF-UB, UR-UL, DF-DL, DR
-DB $/(3,3)/(0,3)/(1,-5)/(4,-2)/(4,1)/(3,3)/$ [7*|18*]
- 4  Swap UF-UB, DF-DB $(1,0)/(5,-1)/(-5,1)/(5,0)$ [3*|9*]
- 5  Swap UR-UB, DR-DB $(0,2)/(0,-3)/(1,1)/(-1,2)/(0,-2)$ [4*|11*]
- 6  Swap UF-UB, DR-DB $(1,0)/(0,-1)/(0,-3)/(5,0)/(-5,0)/(0,3)/(0,1)/($
5,0) [7*|15*]
- 7  Swap UF-UB, UL-UR, DF-DB $/(3,3)/(0,-3)/(3,3)/(0,-2)/(-2,4)/(2,-4)/(0,-1$
)/(3,3)/(0,3) [9*|23*]

b. The following sequences move top layer edges only:

- 1  Swap UF-UB, UR
-UL $/(3,-3)/(3,-3)/(0,1)/(-3,3)/(-3,3)/(-1,0)$ [6*|16*]
- 2  Swap UF-UL, UR
-UB $/(3,3)/(0,3)/(1,1)/(-1,-4)/(-3,-3)/$ [6*|15*]
- 3  Cycle UF->UB->
UR $(1,0)/(-1,2)/(-5,1)/(0,3)/(-3,0)/(5,2)/(-5,4)/(-4,0)$ [7*|19]
 $(-3,0)/(3,0)/(1,0)/(0,-3)/(-1,0)/(-3,0)/(1,0)/(0,3)/(2,0)$ [8*|17*]

4		Swap UF-UB	$(3,0)/(3,3)/(-1,0)/(2,-4)/(4,-2)/(0,-2)/(-4,2)/(1,-5)/(3,0)/(3,3)/$	[10* 26*]
5		Swap UF-UR	$(0,3)/(-3,-3)/(0,-5)/(-2,2)/(0,4)/(0,-4)/(-1,2)/(3,0)/(3,4)/(0,-4)/(0,1)/(3,0)/(0,3)/(-3,0)/(0,3)/(0,-3)/(0,3)/(2,0)/(0,2)/(-2,0)/(4,0)/(0,-2)/(0,2)/(-1,4)/(0,-3)/$	[12* 28*] [13 27*]
6		Cycle UF->UL->UB->UR	$(1,0)/(2,2)/(0,-2)/(3,3)/(1,0)/(4,4)/(0,-2)/(2,2)/(0,-1)/(3,3)/$	[10* 25*]
7		Cycle UB->UF->UL->UR	$(-3,2)/(1,-2)/(-4,0)/(0,3)/(1,0)/(3,-2)/(-4,0)/(-4,0)/(-2,2)/(-1,0)/(0,-3)/$	[11* 26*]

Phase 4: Fix the central layer.

- a. If the middle layer is not a square, then the following sequence will correct it by reversing its right hand side:
 $/(6,0)/(6,0)/(6,0)$

Solution 3: Edges First.

First solve the edges completely, in the same way as phase 1 of method 2. Then put the corners in their layers and solve them using the sequences in the tables below:

- a. The following move corners from one layer to the other.
- Swap UFR-DFR, UBR-DBR, UBL-DBL, UFL-DFL: $(4,0)/(2,2)/(-3,3)/(-2,-2)/(-1,-3)$ [4*|13*]
 - Swap UFL-DFL, UBR-DBR: $(4,0)/(2,2)/(6,-2)$ [2*|7*]
 - Swap UFL-DBR, UBR-DFL: $(0,2)/(-2,-2)/(2,0)$ [2*|6*]
 - Swap UFL-DFL, UFR-DFR: $(1,0)/(3,0)/(2,2)/(-5,-2)/(-1,0)$ [4*|11*]
 - Swap UFL-DFR, UFR-DFL: $(1,0)/(3,0)/(-4,2)/(4,-2)/(3,0)/(-1,0)$ [5*|13*], or $/(3,0)/(-5,0)/(2,2)/(3,-2)/(-3,0)/$ [6|13*]
 - Swap UFL-DFL, UBR-DFR: $/(3,0)/(0,-4)/(6,-2)/(0,-3)/(4,0)/(2,6)$ [6*|14*]
 - Swap UFL-DFR, UBR-DFL: $(0,6)/(0,3)/(0,-4)/(4,0)/(-3,0)/(6,-2)/(2,0)$ [6*|14*]
 - Swap UFR-UBR, UFL-DFR: $(0,2)/(0,-3)/(-2,1)/(2,2)/(0,-3)/(0,3)/(0,-2)$ [6*|15*]
 - Swap UFL-UBR, UFR-DFR: $(0,2)/(0,-3)/(-2,-5)/(2,2)/(0,3)/(0,3)/(0,-2)$ [6*|15*]
 - Swap UFL-UFR, UBR-DFR: $(-2,0)/(0,3)/(0,-3)/(2,2)/(-2,1)/(0,-3)/(2,0)$ [6*|15*]
 - Cycle UFL->UFR->DFR: $(4,5)/(3,0)/(-4,0)/(0,4)/(0,-3)/(0,-4)/(0,-5)$ [6*|14*]
 - Cycle UFL->UBR->DFR: $(-2,0)/(2,0)/(0,-3)/(4,0)/(-4,0)/(0,3)/(-2,0)/(-4,0)$ [7*|15*]
 - Swap UFR-DFR: $/(3,3)/(-1,0)/(2,-2)/(0,-1)/(0,3)/(-2,-1)/(2,0)/(-1,-2)/(-3,0)/$ [10*|23*]
- b. The following move corners within both layers.
- Swap UFR-UBL, UFL-UBR, DFR-DBL, DFL-DBR:

- (1,0)/(-1,5)/(3,3)/(1,1)/(-3,3)/(5,0) [5*|15*]
2. Swap UFR-UFL, UBL-UBR, DFR-DFL, DBL-DBR:
(1,0)/(0,-3)/(2,2)/(-3,3)/(-2,-2)/(3,0)/(-1,0) [6*|16*]
3. Swap UFL-UBR, DFL-DBR: (0,5)/(-2,4)/(2,-4)/(0,1) [3*|9*]
4. Swap UFL-UFR, DFL-DFR: (1,0)/(-3,0)/(2,2)/(1,-2)/(-1,0) [4*|11*]
5. Swap UFL-UBR, DFL-DFR: (4,0)/(0,2)/(0,3)/(-4,0)/(4,0)/(0,-3)/(0,-2)/(2,0) [7*|15*]
6. Swap UFL-UBR, UFR-UBL, DFL-DBR: /(3,3)/(1,2)/(2,-4)/(4,-2)/(-2,-1)/(3,3)/ [7*|19*]
- c. The following move corners in the top layer only.
1. Swap UFL-UBR, UFR-UBL: /(3,-3)/(3,-3)/(0,-2)/(-3,3)/(-3,3)/(2,0) [6*|16*]
2. Swap UFL-UFR, UBR-UBL: /(3,3)/(-5,-2)/(2,2)/(3,0)/(-3,-3)/ [6*|15*]
3. Cycle UFL->UBR->UFR: (1,0)/(0,3)/(2,0)/(-3,0)/(-2,0)/(0,-3)/(2,0)/(3,0)/(-3,0) [8*|17*]
4. Swap UFL-UBR: /(3,3)/(3,0)/(-4,2)/(-2,4)/(6,2)/(-4,2)/(-2,4)/(-5,0)/(-3,-3)/ [10*|26], or
/(-3,0)/(0,-3)/(2,0)/(0,2)/(-2,0)/(4,0)/(0,-2)/(0,2)/(-4,1)/(3,0)/ [11|22*]
5. Swap UFL-UFR: /(3,3)/(1,4)/(0,4)/(-2,4)/(-4,2)/(2,0)/(-2,4)/(4,1)/(3,3)/ [10*|26], or
/(0,3)/(1,2)/(0,-4)/(2,-2)/(0,-4)/(0,-4)/(0,4)/(0,-4)/(-3,0)/(0,-3)/ [11|23*]
6. Cycle UFL->UBL->UBR->UFR:
/(3,3)/(-1,0)/(-2,-2)/(2,0)/(2,2)/(0,-2)/(2,2)/(-1,0)/(-3,-3)/ [10*|24*]

Solution 4: Layer method

Phase 1: Put pieces in their layers.

This phase puts all the white pieces in the top layer, and all the green pieces in the bottom layer.

You can always keep the top and bottom layers square throughout this phase.

- a. First put all the green corners in the bottom layer. This is quite easy if you first put two green ones in the left half of the bottom layer, then pair up the other two green corners and also twist them into the bottom layer.
- b. Now bring the edges to their layer. There are 8 possible cases, which are listed below. In each case I will list the positions where there are edges in the wrong layer.
1. All edges are in the correct layer: do nothing.
 2. UB, DF: do moves (0,-1)/(-3,0)/(4,1)/(-4,-1)/(3,0)/(0,1) [5*|13], or
(-2,0)/(5,0)/(0,3)/(-5,0)/(5,0)/(0,-3)/ [6|12*]
 3. UF, UB, DF, DB: do moves (-1,0)/(1,1)/(0,-1) [2*|4*]
 4. UB, UR, DL, DB: do moves (1,0)/(0,3)/(-1,-1)/(0,-3)/(0,1) [4*|10*]
 5. UF, UR, DF, DB: do moves (1,0)/(0,-3)/(0,3)/(2,-1)/(-3,0)/(0,1) [5*|12*]
 6. UF, UB, DF, DR: do moves (0,-1)/(3,0)/(-3,0)/(1,-2)/(0,3)/(-1,0) [5*|12*]
 7. UR, UB, UL, DF, DR, DB: do moves (0,-1)/(1,4)/(-1,-4)/(-3,0)/(4,1)/(-1,0) [5*|14*]
 8. All edges wrong: do moves (-1,0)/(1,1)/(2,2)/(1,1)/(0,-1) [4*|12*], or apply sequence 3 twice.

Phase 2: Solve each layer separately.

The following table shows all sequences that solve the top layer for each possible case, except for

the ones already listed before, such as those in method 2, part 3b. The bottom layer can be solved in the same way using the mirror images of these sequences (or by turning over the puzzle first).

- a. The following swap the diagonally opposite corners UFL and UBR, as well as the U layer edges as shown:

1. B-L: (4,0)/(3,0)/(3,3)/(3,0)/(5,0)/(0,3)/(3,3)/(0,3)/ [8*|18*]
2. B-R: (1,0)/(3,0)/(3,3)/(3,0)/(5,0)/(0,3)/(3,3)/(0,3)/(3,0) [8*|19*]
3. L-R: (-3,0)/(3,3)/(3,0)/(3,3)/(3,0)/(3,3)/ [6*|15*]
4. F-B: (1,0)/(3,3)/(3,0)/(3,3)/(3,0)/(3,3)/(2,0) [6*|16*]
5. R->B->F: (-2,-1)/(0,3)/(0,1)/(-4,0)/(4,0)/(4,0)/(0,4)/(-4,0)/(2,1)/(3,-3)/(3,0)/(-1,-2) [11*|27*]
6. R->L->F: /(3,-3)/(0,-3)/(-1,4)/(-4,0)/(0,4)/(4,0)/(4,0)/(-4,0)/(0,1)/(0,3)/(0,6) [11*|24*]
7. B-L, F-R: /(3,3)/(-1,4)/(4,-2)/(2,-4)/(0,2)/(4,-2)/(-5,1)/(3,0)/(3,3)/ [10*|26*]
8. L-F, B-R: /(3,3)/(0,3)/(5,-1)/(-4,2)/(-2,0)/(-2,4)/(-4,2)/(2,1)/(3,3)/ [10*|26*]

- b. The following swap the adjacent corners UBL and UBR, as well as the U layer edges as shown:

9. R-F: (1,0)/(3,3)/(-4,-1)/(1,4)/(0,-3)/(0,-3)/(3,0)/(0,-3)/(-4,0) [8*|20*]
10. B-R: /(-3,3)/(3,0)/(0,-3)/(0,3)/(0,-3)/(3,0) [6*|13*]
11. L-B: (1,0)/(-3,3)/(3,0)/(0,-3)/(0,3)/(0,-3)/(2,0) [6*|14*]
12. F-L: /(-3,-3)/(4,1)/(-4,-1)/(0,3)/(3,0)/(-3,0)/(3,0)/(3,0) [8*|19*]
13. L-R: (0,-1)/(1,1)/(3,0)/(-3,0)/(-1,-4)/(3,0)/(-3,0)/(-2,4)/(3,0)/(2,0) [9*|22*]
14. F-B: (3,0)/(3,0)/(3,3)/(3,0)/(4,0)/(0,3)/(3,3)/(0,3)/(2,0) [8*|19*]
15. R->L->F:
/(0,-3)/(-2,0)/(-3,-1)/(-1,2)/(0,-4)/(0,4)/(1,-4)/(0,3)/(0,-5)/(0,3)/(3,0)/(5,0) [12|27*] or
/(0,3)/(0,-3)/(-5,0)/(0,3)/(0,5)/(0,-4)/(4,0)/(-2,-2)/(0,2)/(0,1)/(3,0)/(0,-1)/(0,4) [13|27*] or
(3,3)/(1,6)/(-2,4)/(0,2)/(0,5)/(0,3)/(-2,-1)/(4,0)/(-4,0)/(3,-2)/(-3,0)/(0,6) [11*|29]
16. B->F->R: /(3,0)/(1,2)/(2,0)/(-1,-2)/(0,3)/(-1,0)/(2,0)/(-2,0)/(-1,0)/(-3,-3)/ [11*|24*]
17. L->R->B:
/(3,0)/(0,2)/(0,-3)/(3,0)/(2,0)/(0,4)/(0,4)/(-4,0)/(-1,0)/(-3,0)/(-3,-4)/(0,4) [12|25*] or
(1,-1)/(0,3)/(2,1)/(0,4)/(4,-4)/(-2,3)/(0,3)/(-2,3)/(0,-4)/(-2,0)/(-3,0)/(5,1) [11*|29]
18. F->B->L:
/(3,0)/(1,0)/(-2,0)/(0,5)/(0,-1)/(4,0)/(0,2)/(-4,1)/(0,1)/(0,5)/(-3,0)/(-1,1) [12|26*] or
(4,-1)/(0,3)/(-3,0)/(1,-1)/(4,0)/(2,-2)/(4,0)/(4,0)/(-4,0)/(1,0)/(0,3)/(5,-2) [11*|27]
19. F-L, B-R: /(3,3)/(0,3)/(1,-1)/(-2,0)/(0,2)/(0,-4)/(2,0)/(-2,0)/(1,0)/(-3,-3)/ [11|24*] or
(-3,-3)/(-3,-3)/(-4,5)/(-4,0)/(4,-2)/(-4,2)/(-2,0)/(5,1)/(-3,0)/(3,3)/ [10*|27]
20. F-R, B-L: /(3,3)/(-1,2)/(-4,2)/(4,0)/(-2,4)/(-4,2)/(-1,1)/(3,0)/(3,3)/ [10*|26*]
21. L-R, F-B: /(3,3)/(-1,0)/(2,0)/(0,4)/(0,-4)/(2,2)/(0,-4)/(2,0)/(0,1)/(3,3)/(6,0) [11|25*] or
/(3,3)/(1,-2)/(0,4)/(-2,4)/(2,-4)/(-1,1)/(5,-1)/(-2,1)/(3,3)/ [10*|27]
22. F->R->L->B: (-3,0)/(-3,0)/(3,3)/(3,0)/(1,-2)/(-1,2)/(-3,-3)/ [7*|18*]
23. F->L->R->B: (4,0)/(-3,0)/(3,3)/(3,0)/(2,-1)/(-2,1)/(-3,-3)/(-1,0) [7*|19] or
/(-3,3)/(3,-3)/(1,0)/(0,3)/(0,-3)/(0,3)/(0,-3)/(2,0) [8|18*]

Solution 5: Alternative layer method

Phase 1: Put pieces in their layers.

This phase is the same as in Method 4.

Phase 2: Adjust the permutation parities of the layers.

This phase make sure that each layer is in itself an even permutation. This cuts down the number of possibilities, leaving fewer (and generally shorter) sequences in the last phases. It is however slightly tricky to recognize the parities quickly.

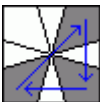
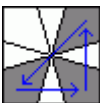
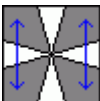
a. Depending on the parities of the layer permutations, do one of the following:

1. Both layers even parity: Do nothing.
2. Top layer odd, bottom even: $/(-3,-3)/(0,-1)/(2,-4)/(4,-2)/(1,0)/(-3,-3)/$
3. Top layer even, bottom odd: $/ (3,3)/(1,0)/(4,-2)/(2,-4)/(0,-1)/(3,3)/$
4. Both layers odd parity: $(1,0)/(5,-1)/(-5,1)/(-1,0)$

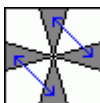
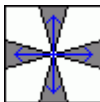
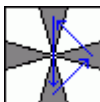
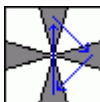
Phase 3: Solve the first layer.

The following table shows the sequences that solve the top layer for each possible case. Note that these sequences may mix up the (unsolved) bottom layer.

a. The following sequences move only corners:

1.  Cycle UFL->UBR->UFR, $(-2,0)/(0,-3)/(2,2)/(0,-3)/(-2,4)/(2,0)$ [5|13*]
1.  Cycle UFL->UFR->UBR, $(-2,0)/(2,-4)/(0,3)/(-2,-2)/(0,3)/(2,0)$ [5|13*]
1.  Swap UFL-UBL, UFR-UBR $/ (3,3)/(1,-2)/(2,2)/(-3,0)/(-3,-3)/$ [6|15*]

b. The following sequences move only edges:

1.  Swap UL-UF, UR-UB $/ (3,3)/(0,3)/(1,1)/(-1,-4)/(-3,-3)/$ [6|15*]
2.  Swap UL-UR, UF-UB $(1,0)/(3,-3)/(2,2)/(3,3)/(4,-2)/(-1,0)$ [5|15*]
3.  Cycle UF->UB->UR $(0,-1)/(0,-3)/(1,1)/(0,-3)/(-1,5)/(0,1)$ [5|13*]
4.  Cycle UF->UR->UB $(0,-1)/(1,-5)/(0,3)/(-1,-1)/(0,3)/(0,1)$ [5|13*]

c. The following sequences swaps the UFL,UBL corners and moves edges:

1		Swap UFL-UBL, UL-UF	$(-2,0)/(0,3)/(-3,-3)/(3,0)/(2,0)$	[4 10*]
2		Swap UFL-UBL, UL-UB	$(3,0)/(3,0)/(3,3)/(3,0)/(3,0)$	[4 10*]
3		Swap UFL-UBL, UL-UR	$(-2,0)/(-3,0)/(2,2)/(0,3)/(1,-5)/(2,0)$	[5 13*]
4		Swap UFL-UBL, UB-UR	$(1,0)/(3,3)/(2,-1)/(-2,1)/(-3,-3)/(-1,0)$	[5 15*]
5		Swap UFL-UBL, UR-UF	$/(3,3)/(1,-2)/(-4,5)/(3,3)/$	[5 13*]
6		Swap UFL-UBL, UB-UF	$(1,0)/(2,-1)/(1,-2)/(2,-1)/(-5,4)/(5,0)$	[5 15*]
7		Swap UFL-UBL, Cycle UL->UB->U F->UR	$(-2,0)/(0,-3)/(3,3)/(2,-1)/(3,3)/(4,-2)/(-1,0)$	[6 17*]
8		Swap UFL-UBL, Cycle UB->UR->U L->UF	$/(3,3)/(1,4)/(2,-4)/(3,0)/(-3,-3)/$	[6 15*]
9		Swap UFL-UBL, Cycle UR->UF->U B->UL	$(-2,0)/(3,3)/(3,0)/(5,-1)/(1,-2)/(3,3)/(2,0)$	[6 17*]
10		Swap UFL-UBL, Cycle UF->UL->U R->UB	$/(3,3)/(0,3)/(1,-5)/(-4,-1)/(3,3)/$	[6 15*]

d. The following sequences swaps the UFL,UBR corners and moves edges:

1.		Swap UFL-UBR, UR-UF	$(-2,0)/(2,-1)/(4,-5)/(-1,2)/(4,-5)/(2,0)$	[5 15*]
2.		Swap UFL-UBR, UB-UF	$(1,0)/(-3,3)/(3,-3)/(-1,0)$	[3 9*]

- 3.

Phase 4: Solve the second layer.

Turn the puzzle over, and solve the top layer using the sequences in the table below. This table is very similar to the one in the previous phase, except that the move sequences are generally a bit longer, because these have to keep the bottom layer intact.

a. The following sequences move only corners:

1.

3.

5.

b. The following sequences move only edges:

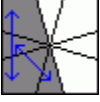
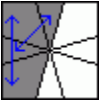
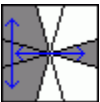
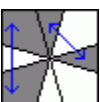
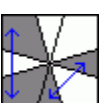
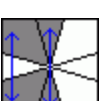
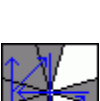
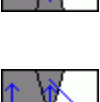
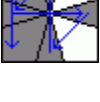
1.

3.

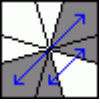
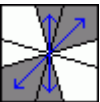
5.

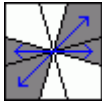
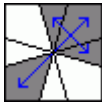
7.

c. The following sequences swaps the UFL,UBL corners and moves edges:

- 1  Swap UFL-UBL, UL-UF $(1,0)/(0,-3)/(0,3)/(0,-3)/(3,0)/(-3,3)/(2,0)$ [6*|14*]
- 2  Swap UFL-UBL, UL-UB $(3,0)/(-3,3)/(3,0)/(0,-3)/(0,3)/(0,-3)/$ [6*|13*]
- 3  Swap UFL-UBL, UL-UR $/(0,3)/(3,3)/(0,3)/(4,0)/(3,0)/(3,3)/(3,0)/(5,0)$ [8*|18*]
- 4  Swap UFL-UBL, UB-UR $(1,0)/(3,0)/(-3,0)/(3,0)/(0,3)/(2,5)/(1,4)/(3,3)/(2,0)$ [8*|20*]
- 5  Swap UFL-UBL, UR-UF $/(3,3)/(1,-2)/(-4,-1)/(0,3)/(3,0)/(-3,0)/(3,0)/(0,-3)$ [8*|19*]
- 6  Swap UFL-UBL, UB-UF $(-2,0)/(-1,-1)/(-3,0)/(3,0)/(1,4)/(-3,0)/(3,0)/(2,-4)/(-3,0)/(0,1)$ [9*|22*]
- 7  Swap UFL-UBL, Cycle UL->UB->UF->UR $(1,0)/(0,3)/(0,-3)/(0,3)/(0,-3)/(2,0)/(-3,3)/(3,-3)/(-2,0)/(3,3)/(2,-1)/(1,-2)/(0,3)/(-3,-3)/(-3,0)/(-1,0)$ [8|18*]
[7*|19]
- 8  Swap UFL-UBL, Cycle UB->UL->UF->UR $/(3,3)/(1,4)/(2,-1)/(3,0)/(-3,-3)/(0,-3)/(0,-3)/(-3,3)/(3,-3)/(-2,0)/(0,3)/(0,-3)/(0,3)/(0,-3)/(-1,0)$ [7*|18*]
- 9  Swap UFL-UBL, Cycle UR->UL->UF->UB->UL $(1,0)/(3,0)/(3,3)/(0,-3)/(-1,2)/(-2,1)/(-3,-3)/(2,0)$ [8|18*]
[7*|19]
- 10  Swap UFL-UBL, Cycle UF->UL->UR->UB $/(-3,0)/(3,3)/(3,0)/(1,-2)/(-1,2)/(-3,-3)/(-3,0)$ [7*|18*]

d. The following sequences swaps the UFL,UBR corners and moves edges:

- 1  Swap UFL-UBR, UR-UF $(1,0)/(2,2)/(3,0)/(0,-3)/(4,1)/(3,0)/(-3,0)/(5,-1)/(0,3)/(0,-2)$ [9*|22*]
- 2  Swap UFL-UBR, UB-UF $(1,0)/(3,3)/(3,0)/(3,3)/(3,0)/(3,3)/(2,0)$ [6*|16*]

- 3  Swap UFL-UBR, UL
-UR $/(3,3)/(3,0)/(3,3)/(3,0)/(3,3)/(3,0)$ [6*|15*]
- 4  Swap UFL-UBR, UB
-UR $(1,0)/(3,0)/(3,3)/(3,0)/(5,0)/(0,3)/(3,3)/(0,3)/(3,0)$ [8*|19*]

If you find that you have erred, and have a last layer with an odd permutation parity, the following move sequence will fix it while keeping the bottom layer intact.

$/(-3,0)/(0,-3)/(2,0)/(0,2)/(-2,0)/(4,0)/(0,-2)/(0,2)/(-4,1)/(3,0)/$ [11|22*]

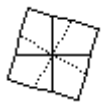
Section 5 SSQ1

The Super Square One is a 4-layer version of the Square One. Just like the Square One, it can adopt non-cubic shapes as it is twisted. As of 2009, it is sold by Uwe Mèffert in his puzzle shop, Meffert's.

Despite its appearance, the Super Square One is not significantly more difficult to solve than the original Square One. The middle layers are nearly identical to the top and bottom layers of the Square One, and may be solved independently using the same methods as the Square One. The corners of the middle layers are distinguishable because the corners with the same two colors are mirror images of each other, but the middle edges of each face are interchangeable since they show only one color each.

解法同 SQ-1，中间两层和顶底两层可采用不同步骤。

中间两层，相对位置判断比较麻烦，首先要把角块还原，用作参照。简单方法：1.复形 2.角块还原（当做一个 2X2X2 魔方处理） 3.楞块回层 4.楞块位置。



楞块回层需要两个公式：



Swap DF-UB, DB-UF: $(0,-1)/(1,1)/(-1,0)$

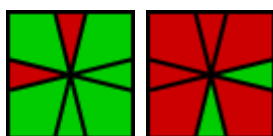


Swap DR-UR, UF-UB: $(1,0)/(0,-3)/(-1,-4)/(1,1)/(0,3)/(0,3)/(-1,0)$

第二个公式只调换下层一个棱块到上层，只用这个公式就可以把所有楞块归层。

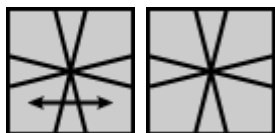
顶底两层判断方便，采用 Lars Vandenberg 的方法比较方便：1.复形 2.角块归层 3.楞块归层 4.置换角块位置 5.置换楞块位置

楞块归层只需要一个公式，对调 DB-UL, DR-UB

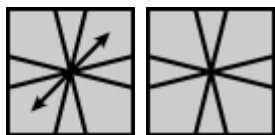


$(1,0) / (-3,0) / (-1,-1) / (4,1) / (-1,0)$

置换角块位置两个公式：

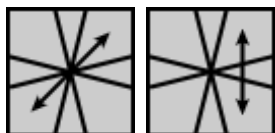


$/ (3,-3) / (3,0) / (-3,0) / (0,3) / (-3,0) /$



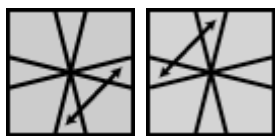
$/ (3,3) / (3,0) / (3,3) / (3,0) / (3,3) /$

也可以只用下面一个公式



$/ (3,0) / (-3,0) / (3,0) / (-3,0) /$

置换楞块位置，只用一个公式：

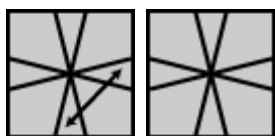


(UR UF) (DL DF)

$(1,0) / (0,3) / (-1,-1) / (1,-2) / (-1,0)$

对调相邻楞块，先完成一层，另一层的两个楞块对换当做辅助。下一层利用这个公式偶数次即可。

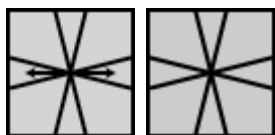
有时只剩一对棱需要对换，用下面公式



(UR UF)

$/ (-3,0) / (0,3) / (0,-3) / (0,3) / (2,0) / (0,2) / (-2,0) / (4,0) / (0,-2) / (0,2) / (-1,4) / (0,-3)$
 $/ (0,3)$

[13|27]



(UR UL)

$/ (3,3) / (-1,0) / (2,-4) / (4,-2) / (0,-2) / (-4,2) / (1,-5) / (3,0) / (3,3) / (3,0)$

[10|26]

第三十节 Skewb

The **Skewb** is invented by Tony Durham and marketed by Uwe Mèffert. it differs from Rubik's construction in that its axes of rotation pass through the corners of the cube rather than the centres of the faces. There are four such axes, one for each space diagonal of the cube. As a result, it is a *deep-cut* puzzle in which each twist affects all six faces.

Skewb Diamond is the dual polyhedron of the Skewb. Higher order Skewbs, named **Master Skewb** and **Elite Skewb**, have also been made.

The number of positions of the standard Skewb:

There are 4 fixed corners with 3 orientations, 4 free corners with 3 orientations and 6 face pieces giving a maximum of $6! \cdot 4! \cdot 3^8$ positions. This limit is not reached because

- The total twist of the corners is fixed (3)
- The faces must have an even permutation (2)
- The free corners must have an even permutation, and hence form a tetrad (2)
- The orientations of the fixed corners and the position of one of the free corners will determine the positions of the other three (3)

Another way of describing the last condition is that the relative position of the two tetrads also defines a kind of twist, which must be equal to the twist of the fixed tetrad. This leaves $6! \cdot 4! \cdot 3^6 / 4 = 3,149,280$ positions.

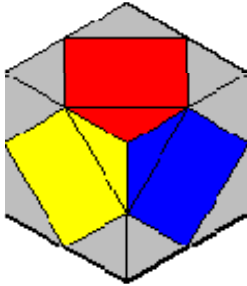
Every position can be solved in at most 11 moves.

Blake O'Hare's solution(<http://www.acubist.com/>)

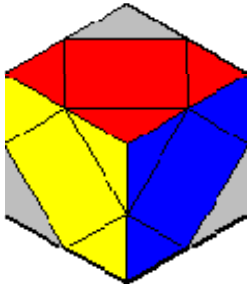


Section 1: Beginner's solution

Step 1: Placing Three Centers



Step 2: Placing Three Corners



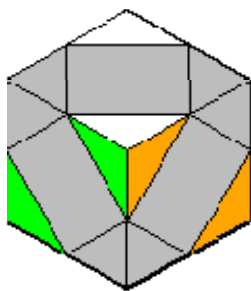
这一步是复原 LRD 三个角如下图：



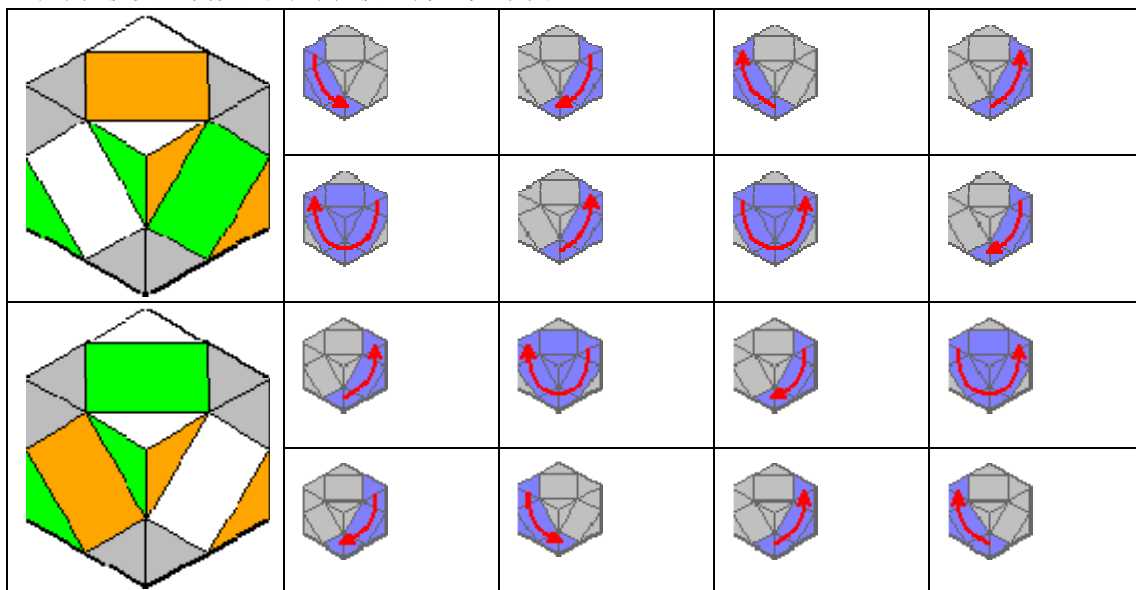
首先复原两个角块，那么剩下的一个角块必定在正确位置，方向需要顺时针或逆时针扭转。
针对这两种情况这个角块复原方法如下图所示：

Step 3 : Last Three Centers

将 Skewb 翻转过来，还原另一半，下图灰色块：



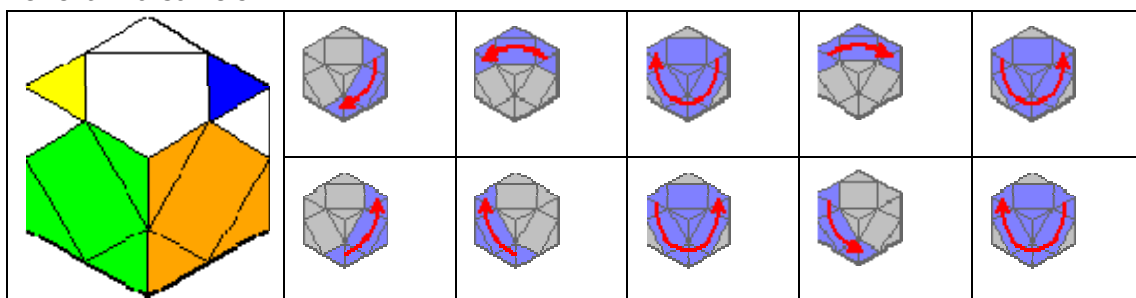
分两步走，这一步还原三个中心块，第四步还原剩下三个角块。三个中心块位置要么需要顺时针轮换要么需要逆时针轮换，方法如下图：



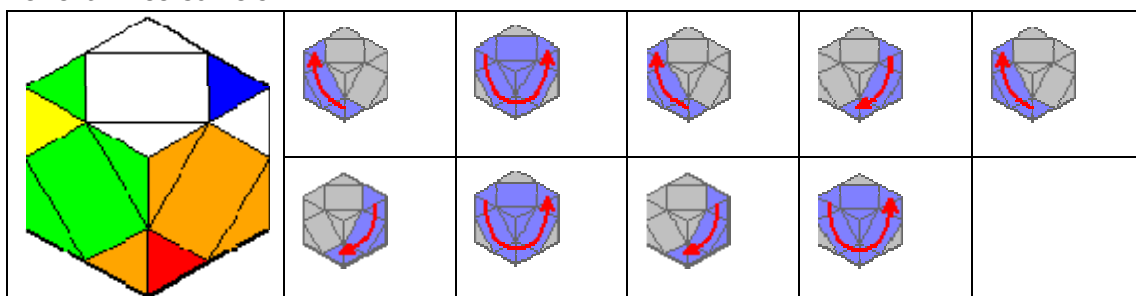
Step 4: Last Three Corners

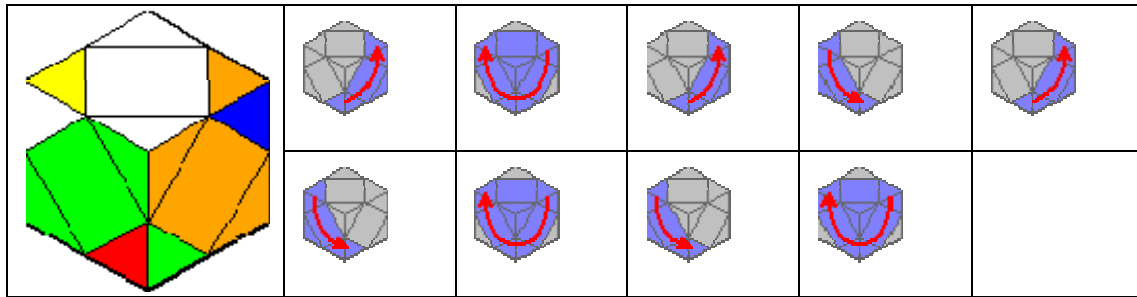
Having only two misoriented corners is what happens most (67%) of the time. The three corner situation happens 22% of the time, and the remaining 11% we don't need to do anything at all.

Orient Two Corners



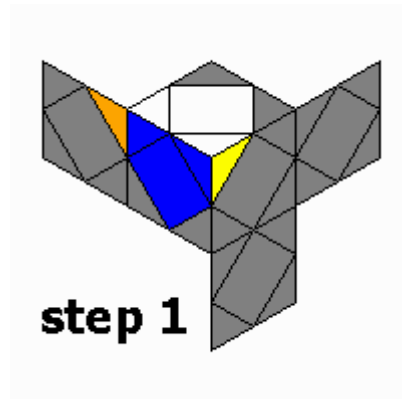
Orient Three Corners



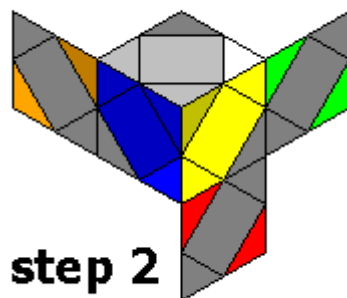


Section 2: Advanced Solution

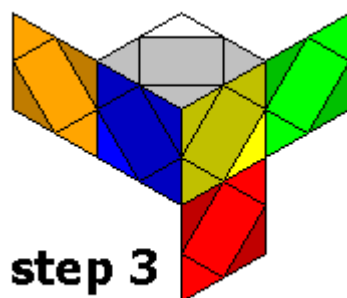
Step 1: Build 2 adjacent centers and the corners in between



Step 2: Build another center and the 2 adjacent corners to finish the first 'half'

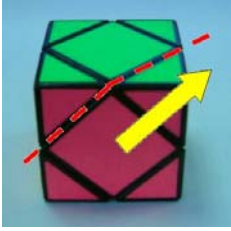
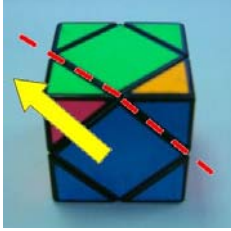
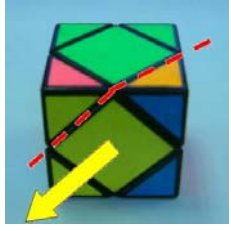
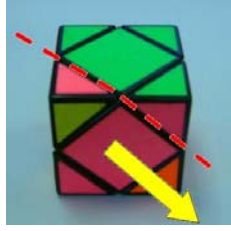


Step 3: Orient the last 3 corners and permute the last 3 centers



Section 3: 许技江 Skewb 解法

一个基本公式：

			
右上	左上	右下	左下

Step 1: 完成顶层

Step 2: 完成底层四个角块

觀察下面四個角中，哪兩個要順時針轉，哪兩個要逆時針轉。把要順時針轉的兩個角，放到面朝你的那面下方（原本轉好的那一面仍然維持在上方），要逆時針轉的兩個角就會在背對你的後面下方。用一次「上上下下」公式，角就會轉好。

特例是：如果下面四個角中，有兩個角已經轉好，另外兩個角一順一逆怎麼辦？

只要記得：用一次「上上下下」公式，讓兩個已經好的角要轉掉，另外兩個角也不要轉好，這樣四個角就可以轉個方向，再套用一次「上上下下」公式。

Step 3: 调整剩余中心块

首先，若要換的三個中心，剛好連成一直線。發現 A 中心要到 B、B 中心要到 C、C 中心要到 A。那請面對 C 中心（位在中間的中心，要去的地方），把 B 中心（位在中間的中心）放在下方，這樣 A 中心是在背面。作一次「上上下下」公式，然後朝上面保持朝上，整個方塊水平旋轉 180 度，面朝背面，再作一次「上上下下」公式，這樣三個面就會對調完成。

比較不一樣的，如果這三個面是形成三角形呢？同樣是 A 中心要到 B、B 中心要到 C、C 中心要到 A。則面對 C 中心，把要去 C 的 B 中心放在下方，然後先轉一下，把 A 中心轉到後面去；再如上述調動三中心後，把後面的中心再轉回來。

另外補充一種：前面和下面中心交換、左面和右面中心交換。只要連續作三次「上上下下」公式即可。

第三十一节 Snake Cube



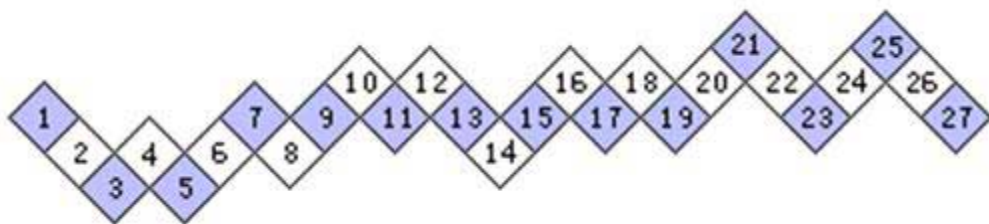
又叫神龙摆尾，蛇形魔方。可以完全用推理推导出来。下面我介绍一下我的玩法，它的关键之处在于找到哪一块应该放到立方体正中心，其后，照直觉顺手就折成立方体。

神龙摆尾拼合起来是个立方体，共有 27 个方块，一般是绿白相间的方块，数数绿色方块和白色方块分别 14，13 个。你看看绿色块和白色块无论是拉开还是拼合好，都必定是相邻的。拼成立方体后有三层，这两种颜色的方块在这三层分别为 5，4，5 共 14 块；4，5，4 共 13 块。而正中心的那块属于 4，5，4 即 13 块这组。如下图这个拼好的立方体，绿色的方块在上中下三层分别为 5，4，5 块共 14 块，白色则应该 13 块。

如下图这个打乱的，数一数白色块有 13 块，绿色（图是论坛里的图，为灰色）块有 14 块，所以，拼好后的立方体，白色方块在上中下三层分别为 4，5，4 个，立方体正中心那块必定是白色块，我假设它是 C 块，现在就是要把 C 块从这 13 个白色块中找出来。

连接 C 块的必定是 6 个面中的中心块，我们叫它 B 块 D 块，因为 BD 是面的中心块，所以它直接相连成直线的必定只有两块，比如 11，12，12 可以是面中心块，而 123 则是三个方块连成直线，所以 2 必定不是面中心块。有了这个约束条件，你会发现只有一块符合这些条件，那就是 6 是中心块 C，5，7 是面心块 B，D。

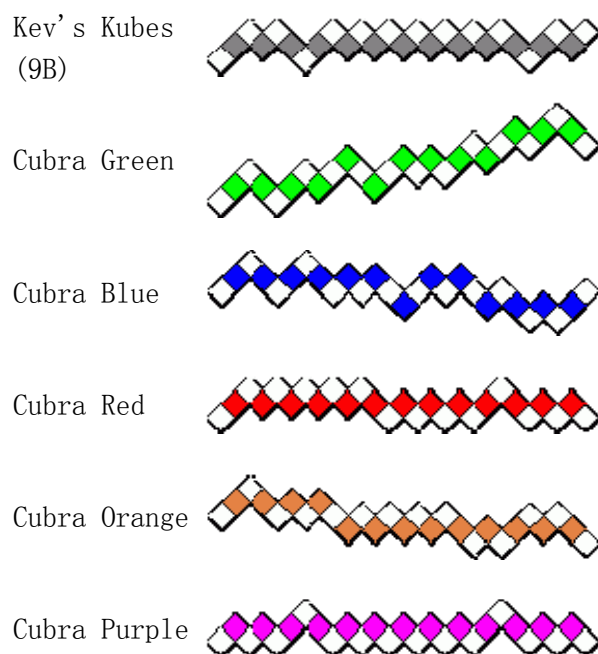
找到了中心剩下的就很简单了，从中心块开始折它相邻的边，只有两种折法，其中一种第二步就就行不通，故而只剩下唯一的一种折法，其后，剩余的边该怎么折合成正方体每一步都只有一种方式，顺着折下去就成了正方体。



There are many versions of this puzzle. Some are made of wood, others of plastic, but more interesting is that different versions have different arrangements of 'straight' and 'bend' cubes on the

string. Some of these arrangements are much easier than others. In general, those with more straights are easier because there are less degrees of freedom. On the other hand, some versions may have multiple solutions making it easier, too. The most difficult ones are those with unique solutions and not too many straights.

One of the hardest versions of the snake cube is Kev's Kubes (version 9B), made by Trench Puzzles. It is made from wood, alternately coloured white and black. A plastic version called Cubra comes in 5 variations, each with a different colour. In order of difficulty, they are Mean Green, Bafflin' Blue, Twist yer 'ead Red, 'Orrible Orange, and Puzzlin' Purple



还有 4x4x4 的版本，叫做 King Snake Cube。



第三十二节 其它异形

Section 1: 镜面魔方(Mirror Cube)



完全等价与普通三阶，不同的块用大小来区分。

Section 2: 空心魔方(Void Cube)

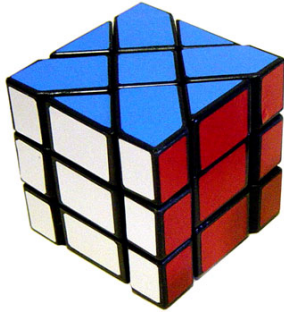


Void Cube 是 2008 年日本的 Okamoto Katsuhiko 设计的。三阶 Rubik's Cube，固定角块楞块，中心块可以有 12 种排列方式（六个面心和内部轴心形成的整体可以有 24 种朝向，但其中十二种是面心的奇排列，从整个魔方角度来看是不会出现的），对于空心魔方来说，这十二种状态是等价的，所以空心魔方的组合数是普通三阶的十二分之一。也可以换一种方式来看，用一个角块来定魔方的整体方向，有 24 种等价情况，是三阶的 $1/24$ ，转动中间层可以看到空心魔方允许奇置换，普通三阶只允许偶置换，是三阶的 2 倍，所以总体组合数是三阶的 $1/12$ 。

还原方法也完全可以按照三阶来。按照某些三阶玩法，最后会出现需要交换一对棱或者一对角块的情形，这实际上等价于三阶的交换一对棱和几个中心块。比如交换 UF，UR 两个楞块可以用公式 $R F U D' L2 B2 L F' B D' B2 D B'$ ，交换 UF，UB: $R2 F U D' L' F2 R B2 L F' B D' F B R'$ ，交换邻角 LUF，RUF: $U R D' L2 B2 R' U F' U' R B2 L' D R' D' L'$ ，交换对角 LUF，RUB: $R D F U L' B2 D' L' F' R2 B' U R2 F2 D2 B2 D' F2$ 。

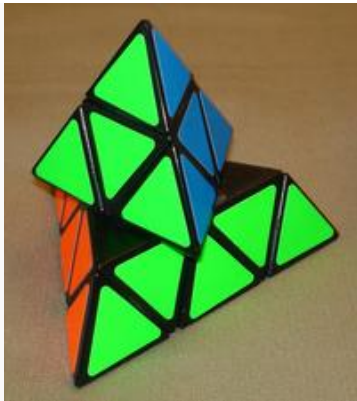
如果用角先，Thistlethwaite 等玩法则不会出现这种奇偶性问题。

Section 3: 移棱魔方(Fisher Cube)



结构跟普通三阶完全一样，外形切割不同。按三阶的玩法即可。

Section 4: 金字塔魔方(Pyraminx)



The Pyraminx is a tetrahedral puzzle similar to the Rubik's Cube. It was invented and patented by Uwe Meffert, and introduced by Tomy Toys of Japan (then the 3rd largest toy company in the world) in 1981.

The Pyraminx is in the shape of a tetrahedron, divided into 4 axial pieces, 6 edge pieces, and 4 trivial tips.

四个顶角块和四个轴块可以作为一个整体，有 3^4 种方向，6 个边块有 $6!/2$ 种位置排列和 2^5 种朝向，总共是 933,120 种组合状态。假如把角块方向调整好（很简单的旋转），只剩下六个楞块的组合 11, 520 种情形。可以用三循环还原。

Section 5: 二阶异形(Pyramorphix)



The Pyramorphix is a tetrahedral puzzle similar to the Rubik's Cube. It has a total of 8 movable

pieces to rearrange. Though it looks like a simpler version of the Pyraminx, the mechanism is identical to that of the Pocket Cube.

它的内部结构和二阶魔方完全一样，用二阶的还原方法复原即可。因为单贴片的那个块没有朝向，所以 Pyramorphix 只有 $8! \times 3^4 / 24$ 即 136, 080 种组合状态，比二阶要少 3^4 倍。

Section 6: Pyraminx Crystal



The Pyraminx Crystal is a dodecahedral puzzle similar to the Rubik's cube and the Megaminx. It is manufactured and sold by Uwe Mèffert in his puzzle shop since 2008.

There are 30 edge pieces with 2 orientations each, and 20 corner pieces with 3 orientations each, giving a maximum of $30! \cdot 2^{29} \cdot 20! \cdot 3^{19}$ possible combinations.

角块，楞块分别只能偶置换，除以 4；总体没有固定方向，可用一个角块来定朝向，除以 60。所以实际的组合数为：

$$30! \times 2^{29} \times 20! \times 3^{19} / 240 = 1.68 \times 10^{66}$$

Section 7: Skewb Ultimate



The Skewb Ultimate, originally marketed as Pyraminx Ball is a twelve-sided puzzle derivation of the Skewb, produced by famous toy-maker Uwe Meffert.

The Skewb Ultimate has 6 large "edge" pieces and 8 smaller corner pieces.

Section 8: Skewb Diamond



The Skewb Diamond is an octahedron-shaped puzzle similar to the Rubik's Cube. It has 14 movable pieces which can be rearranged in a total of 138,240 possible combinations. This puzzle is the dual polyhedron of the Skewb.

The Skewb Diamond has 6 octahedral corner pieces and 8 triangular face centers. All pieces can move relative to each other.

附录 OLL/PLL complete Algorithms

Cited from Wiki

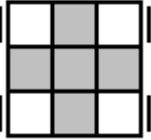
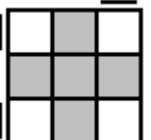
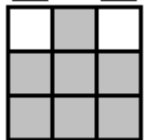
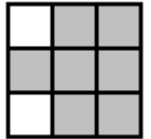
Section 1 OLL

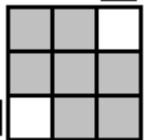
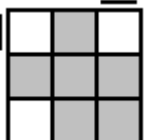
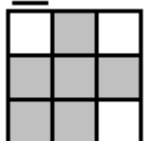
OLL is the Orientation of the Last Layer, and is the first step in solving the last layer in many speedsolving methods, such as the Fridrich Method.

Gray square represent the oriented pieces on the top.

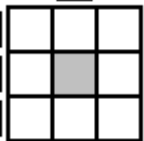
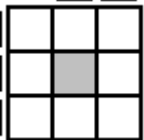
All edges flipped correctly

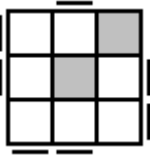
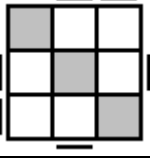
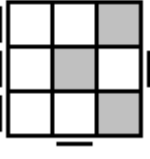
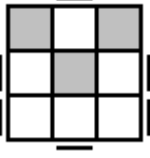
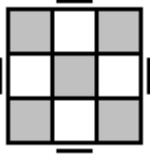
This subgroup, OCLL is used in methods which have oriented the LL edges earlier in the solve, such as ZZ or VH.

OLL 21 	R U R' U R U' R' U R U2 R' y R U2 R' U' R U R' U' R U' R' y F (R U R' U') (R U R' U') (R U R' U') F' R' U' R U' R' U R U' R' U2 R
OLL 22 	R U2 R2' U' R2 U' R2' U2 R f (R U R' U') f' F (R U R' U') F' y2 L' U2 L2 U L2 U L2 U2 L' R U2 R2 U' R' U R U' R U' R' y M' U2 M R' U L U' R U' L' U' L U' L' y2 L U' R' U L' U R U R' U R y2 L' U R U' L U' R' U' R U' R' R U' L' U R' U L U L' U L
OLL 23 	R2 D' R U2 R' D R U2 R R' U2 R F U' R' U' R U F' I2 U' R D2 R' U R D2 R I2' U' z' U R2 U' L U R2 z' L y' R U R' U' (R U' R' U2)(R U' R' U2) R U R' y2 R' F2 R U2 R U2 R' F2 R U2 R' y2 R2 D R' U2 R D' R' U2 R' (R' U' R U) R' U2 R2 U (R' U R U2 R)
OLL 24 	r U R' U' L' U R U' x' r U R' U' r' F R F' r U R' U' L' U R U' y' R U R2' U' R2 U' R2' U2' R U' R U' R' y' R U' R' (U2 R U R') (U2 R U R') (U R U' R') y2 R' F' r U R U' r' F y2 l' U' L U R U' r' F
OLL 25	y2 R' F R B' R' F' R B

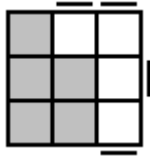
	$y' F' r U R' U' r' F R$ $y' x U R' U' L U R U' r'$ $y R' F' L' F R F' L F$ $y R' U' (R U R' F') (R U R' U') (R' F R^2)$ $y^2 R U^2' R D R' U^2 R D' R^2'$ $y^2 F R' F' r U R U' r'$ $y^2 l' U R D' R' U' R D x'$ $y^2 x' L' U' R' U L U' R U$ $x' D R U R' D' R U' l'$ $x' U L' U' R U L U' l'$ $l' U' L' U R U' L U x'$ $U r U R U' L' U R' U' x'$ $R U^2 R' U' R U R' U' R U R' U' R U' R'$ $R U R' U R U' R' U R U' R' U R U^2 R'$
OLL 26 	$y^2 R' U' R U' R' U^2 R$ $y' R U^2 R' U' R U' R'$ $L U^2 L' U' L U' L'$ $L' U' L U' L' U^2 L$ $L' U R U' L U R'$
OLL 27 	$R U R' U R U^2 R'$ $y' R' U^2 R U R' U R$ $y L' U^2 L U L' U L$ $y^2 L U L' U L U^2 L'$ $R U' L' U R' U' L$

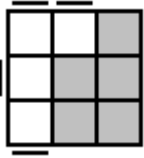
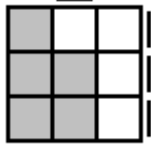
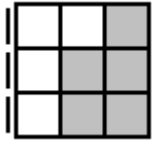
No edges flipped correctly

OLL 1 	$(R U^2) (R^2' F R F' U^2') (R' F R F')$ $R U B' R B R^2 U' R' F R F'$
OLL 2 	$F (R U R' U') F' f (R U R' U') f'$
OLL 3 	$y f (R U R' U') f' U' F (R U R' U') F'$ $y^2 r' R^2 U R' U r U^2 r' U R' r$

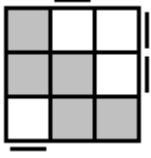
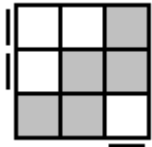
OLL 4 	f (R U R' U') f' U F (R U R' U') F' R' U2 x R' U R U' y R' U' R' U R' F y r' R U' r U2 r' U' R U' R2' r
OLL 17 	R U R' U R' F R F' U2 R' F R F'
OLL 18 	y' r U R' U R U2 r2' U' R U' R' U2 r y2 F R U R' d R' U2 R' F R F' R U2 R2 F R F' U2 M' U R U' r'
OLL 19 	r' U2 R U R' U r2 U2' R' U' R U' r' R' U2 F R U R' U' y' R2 U2 R B r' R U R U R' U' r R2' F R F' M U R U R' U' r R2' F R F'
OLL 20 	r' R U R U R' U' r2 R2' U R U' r' M U R U R' U' M2 U R U' r' M' U2 M' U2 M' U M U2 M U2 M M' U2 M U2 M' U M U2 M' U2 M M d r2 U M' U2 M U r2 S' M' U M' U M' U M' U M U M U M U (r U R' U') M2 (U R U' R') U' M'

P-Shapes

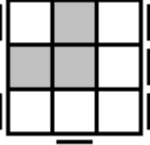
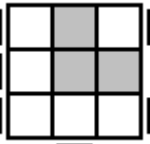
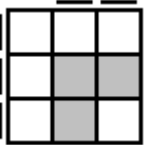
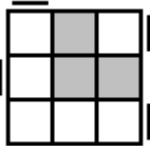
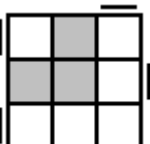
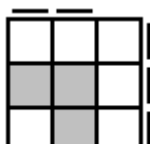
OLL 31 	▪ L' d' R d L U' L' B' L ▪ S' L' U' L U L F' L' f ▪ y x U R' U' l U R U R' U' R U' R' ▪ y2 R' U' F U R U' R' F' R ▪ L' U' B U L U' L' B' L
--	--

OLL 32 	▪ $R d L' d' R' U R B R'$ ▪ $S R U R' U' R' F R f'$ ▪ $y x' U' R U l' U' R' U' R U R' U R$ ▪ $R U B' U' R' U R B R'$ ▪ $y^2 F U R U' F' r U R' U' r'$
OLL 43 	▪ $f L' U' L U f$ ▪ $B' U' R' U R B$ ▪ $y^2 F' U' L' U L F$ ▪ $U R U R' F' L' U' L F$ ▪ $y^2 R' U' F R' F' R U R$
OLL 44 	▪ $f R U R' U' f'$ ▪ $y^2 F U R U' R' F'$ ▪ $y R U y' L U' L' B'$ ▪ $y F U R' F R F' U' F'$

W-Shapes

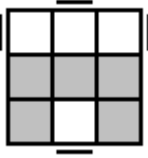
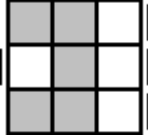
OLL 36 	▪ $R' U' R U' R' U R U R B' R' B$ ▪ $y^2 (L' U' L U') (L' U' L U) (L F' L' F)$ ▪ $(R U R' U') y (L' U^2 L U) y' (R U R')$
OLL 38 	▪ $L U L' U L U' L' U' L' B L B'$ ▪ $y^2 R U R' U R U' R' U' R' F R F'$ ▪ $(L' U' L U) y' (R U^2 R' U') y (L' U' L)$

L-Shapes

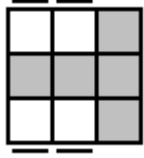
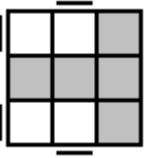
OLL 53 	▪ $F' U' L U' L' U L U' L' U_2 I$ ▪ $y_2 r' U' R U' R' U R U' R' U_2 r$ ▪ $y' r' U_2 R U R' U' R U R' U r$
OLL 54 	▪ $r U R' U R U' R' U R U_2 r'$ ▪ $y' r U_2 R' U' R U R' U' R U' r'$
OLL 50 	▪ $y_2 R' F R_2 B' R_2 F' R_2 B R'$ ▪ $r' U r_2 U' r_2' U' r_2 U r'$ ▪ $R B' R B R_2' U_2 F R' F' R$ ▪ $y_2 I' U I_2 U' I_2 U' I_2 U I'$ ▪ $r' U_2 R_2' U R_2 U r R U_2 R'$ ▪ $y' [R U_2 R' U' R U' R'] [F R U R' U' F']$ ▪ $(y x') U' r U_2 L' U_2 r' U_2 L U' (x)$
OLL 47 	▪ $F' L' U' L U L' U' L U F$ ▪ $R' U' x R' U R U' R' U R U' x' U R$ ▪ $b' U' R' U R U' R' U R b$
OLL 48 	▪ $F R U R' U' R U R' U' F'$ ▪ $y_2 f U R U' R' U R U' R' f'$
OLL 49 	▪ $R B' R_2 F R_2 B R_2 F' R$ ▪ $I U' I_2' U I_2 U I_2' U' I$ ▪ $y_2 r U' r_2 U r_2 U r_2 U' r$ ▪ $y_2 R' F R' F' R_2 d_2 y' R' F R F'$ ▪ $y_2 r U_2 R_2' U' R_2 U' r' R' U_2 R$ ▪ $(y' x') U I' U_2 R U_2 I U_2 R' U (x)$

C-shapes

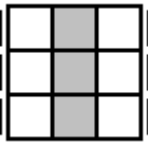
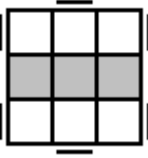
OLL 34	▪ $R U R' U' x D' R' U R E'$ ▪ $y_2 R' U' R U y' x' R U L' U' M'$
---------------	--

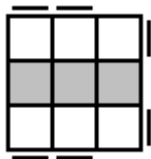
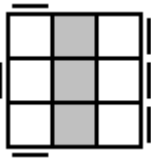
	▪ $RUR'd'l'U'LUR$ ▪ $y^2FRUR'UR'F'rURU'r'$
OLL 46 	▪ $R'U'R'FRF'UR$ ▪ $R'U'l'U'IF'UR$ ▪ $yFUFLw'U'LwU'F'$ ▪ $RUlU'R'U'F'l'$ ▪ $y^2LFU'RUR'F'L'$ ▪ $RUR^2U'R'FRURU'F$

T-Shapes

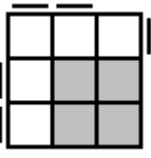
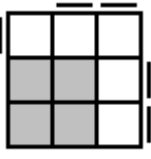
OLL 33 	▪ $RUR'U'R'FRF'$ ▪ $R'U'RUIU'R'U(x)$
OLL 45 	▪ $FRUR'U'F'$ ▪ $Fw'U'L'ULFw$ ▪ $y^2FwURU'R'Fw'$ ▪ $y^2F'L'U'LU'F$

I-Shapes

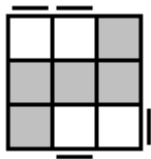
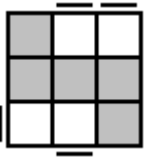
OLL 55 	▪ $RU^2R^2U'RU'R'U^2FRF'$ ▪ $R'U^2R^2UR'URU^2yLw'U'Lw$ ▪ $yR'FRUR'FRU'R'F'RUM'U'M$ ▪ $LF'L'U'LF'L'ULFL'U'M'UM$ ▪ $(LF'L'U'LF'L'ULFL'U'L)(R'FR)L'$ ▪ $Rd'R^2DR'U^2RD'R^2dR'$
OLL 56 	▪ $RwURw'URU'R'URU'R'RwU'Rw'$ ▪ $Rw'U'RwU'R'URU'R'URRw'URw$ ▪ $LF'L'URU'R'URU'R'LF'L'$ ▪ $FRUR'U'RF'RwUR'U'Rw'$ ▪ $RBR'ULU'L'ULU'L'RB'R'$

OLL 51 	▪ $FURU'RURU'R'F'$ ▪ $y2fRUR'U'RUR'U'f'$
OLL 52 	▪ $R'URUR'R'dR'URB$ ▪ $RUR'URd'RUR'F'$ ▪ $RUR'URU2R'(Sune)U(FURU'R'F')$

Square Shapes

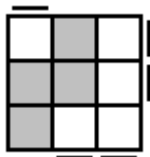
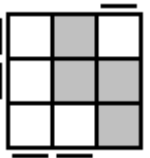
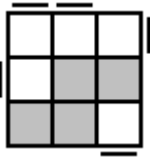
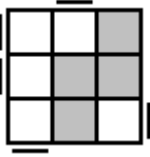
OLL 5 	▪ $(r'U2)(RUR'Ur)$ ▪ $y2l'U2LU'L'U'l'$
OLL 6 	▪ $y2(rU2)(R'U'RU')r'$ ▪ $lU2L'U'L'U'l'$

Big Lightning bolt Shapes

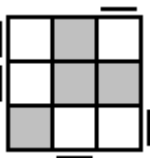
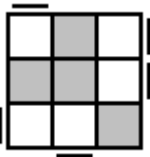
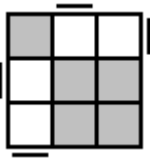
OLL 39 	▪ $LF'L'U'LUFU'L'$ ▪ $y2fLFL'U'L'ULS$ ▪ $y2RB'R'URUBU'R'$
OLL 40 	▪ $R'FRUR'U'F'UR$ ▪ $y2fR'F'RURU'R'S'$ ▪ $y2L'BLUL'U'B'UL$

Small lightning bolt Shapes

OLL 7	▪ $RwUR'URU2Rw'$
--------------	---

	▪ $F' R' F R U^2 R U^2 R'$ ▪ $L' U^2 L U^2 L F' L' F$
OLL 8 	▪ $y^2 r' U' R U' R' U^2 r$ ▪ $R U^2 R' U^2 R' F R F$
OLL 11 	▪ $r' R^2 U R' U R U^2 R' U r R'$ ▪ $y^2 r U R' U R' F R F' R U^2 r'$ ▪ $y F' L' U' L U F' Y F R U R' U' F'$
OLL 12 	▪ $F R U R' U' F' y F R U R' U' F'$ ▪ $y F R U R^2 F R y' R^2 U R U^2 R'$ ▪ $y' r R^2 U' R U' R' U^2 R U' R r'$

Fish Shapes

OLL 9 	▪ $y' R' U' R y' x' R U' R' F R U R' x$ ▪ $y' R U R' U' R' F R^2 U R' U' F'$
OLL 10 	▪ $y' R U R' y R' F R U' R' F' R$ ▪ $y' R' U' R U R B' R^2 U' R U B$
OLL 35 	▪ $R U^2 R^2' F R F' R U^2 R'$ ▪ $y' R' U^2 (R L w) U' R' U L w' U^2 R$
OLL 37	▪ $x' U' R U I' U' R' U R$ ▪ $I U' R' U y' R U R' U' z'$

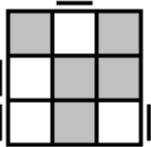
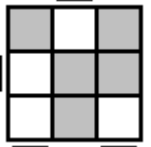
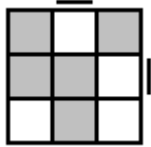
	▪ $yFRU'R'U'RUR'F'$ ▪ $yR'FRF'U'F'UF$ ▪ $yFR'F'RURU'R'$ ▪ $RB'R'BUBU'B'$
--	---

Knight move Shapes

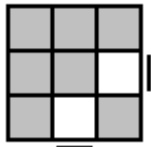
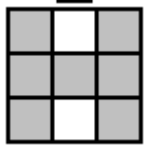
OLL 13 	▪ $FURU'R2F'RURU'R'$ ▪ $FURU2R'U'RUR'F'$ ▪ $y2x'RUR'F'RUR'xyR'UR$ ▪ $y2fRUR2U'R'URU'f'$ ▪ $rU'r'U'rUr'y'R'UR$
OLL 14 	▪ $R'FRUR'F'Ry'RUR'$ ▪ $y'R'U'y'R'UR2BR'U'R'UR$ ▪ $y2B'U'R'UR2BR'U'R'UR$
OLL 15 	▪ $y2R'F'RL'U'LU'R'FR$ ▪ $r'U'rR'U'RUR'rUr$
OLL 16 	▪ $y2rUr'RUR'U'rU'r'$ ▪ $y2rUMUR'U'rU'r'$

Awkward Shapes

OLL 29 	▪ $y2R'FRF'RUR2R'd'L'U'L$ ▪ $y2R2'U'RFR'UR2'U'R'F'R$ ▪ $y(RUR'U')(RU'R'F')U'(FRUR')$ ▪ $yR'U2R2UR'U'R'U2FRF'$ ▪ $MU(RUR'U')(R'FRF')M'$
-------------------	---

OLL 30 	▪ $y^2 F R' F R^2 U' R' U' R U R' F^2$ <---BEAST ▪ $I U' R' U' U^2 d L U L'$ ▪ $f R U R^2 U' R' U R^2 U' R' f$ ▪ $y^2 F U R U' x' U R' D' R U' R'$ ▪ $R^2 U' R' B' R U' R^2 U' R B R'$ ▪ $M U' (L' U' L U) (L F' L' F) M'$
OLL 41 	▪ $R U' R' U^2 R U y R U' R' y' U' R'$ ▪ $y R U^2 R' U' R U' R^2 (y) L' U' L U F$ ▪ $y^2 F U R U' R' F' R' U^2 R U R' U R$ ▪ $y^2 R U R' U R U^2 R' F R U R' U' F'$ ▪ $y^2 R^2 L z' I' U' R d L U' L I'$
OLL 42 	▪ $Rw' R^2 y R U R' U' y' R' U M'$ ▪ $R' U' R U F R U R' U' R' U R U' F'$ ▪ $L' U L U^2 L' U' y' L' U L U F$ ▪ $y R' U^2 R U R' U R^2 y R U R' U' F'$ ▪ $y R' F R F' R' F R F' R U R' U' R U R'$ ▪ $y^2 L^2 R' z R w U L' D w' R' U R' R w$

All Corners Oriented

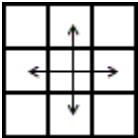
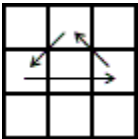
OLL 28 	▪ $Rw U R' U' R w' R U R U' R'$ ▪ $F R U R' U' F' U^2 F R U R' U' F'$ ▪ $y^2 M' U M U^2 M' U M$ ▪ $M U M' U^2 M U M'$
OLL 57 	▪ $R U R' U' R w R' U R U' R w'$ ▪ $(x') M' U' R U M U' R' U (x)$ ▪ $M U M U M U^2 M' U M' U M' U^2$ ▪ $y M' U M' U M' U M' U M' U' M' U' M' U' M' U'$

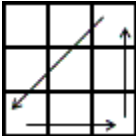
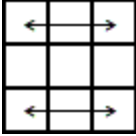
Section 2 PLL

PLL is the Permutation of the Last Layer, the last step of many speedsolving methods. In this step, the pieces on the top layer have already been oriented (OLL) so that the top face has all the same color, and they can now be moved into their solved positions. There are 21 PLLs (13 if you count mirrors and inverses as being the same) and each one is named after a letter. The following page gives a list of all of the PLLs, along with a picture and a list of common algorithms for each one. PLL is a subgroup of ZBLL.

Permutations of Edges Only

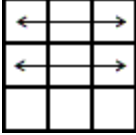
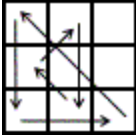
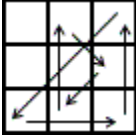
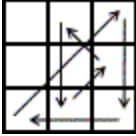
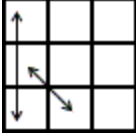
This subgroup, EPLL is used as the last step for many speedsolving methods, for example in the VH method (COLL).

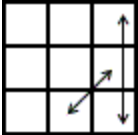
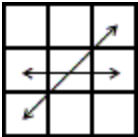
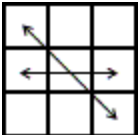
H Permutation 	▪ M2 U' M2 U2' M2 U' M2 ▪ R2' r2 U' L2 l2' U2 R2' r2 U' L2 l2' ▪ U' M2 U2 M2 U M2 U2 M2 ▪ F2 M2 F2 U' F2 M2 F2 U ▪ x U2 M2 U2 B' U2 M2 U2 B ▪ R2 U2 R2 U2 R2 U R2 U2 R2 U2 R2 U' ▪ L R U2 L' R' F' B' U2 F B ▪ R U2 R' U' R' U' R2 U' R2 U2 R2 U2 R' U ▪ R2 U2 R U2 R2 U2 R2 U2 R U2 R2 ▪ M2 U M2 U M2 U M2 U M2 U M2 U' ▪ M2 U M2 U2 M2 U M2 ▪ R' D' R D' B D B' D R' D' R U R' D R D' B D' B' D R' D R U
U Permutation : a 	▪ (y2) F2 U' L R' F2 L' R U' F2 ▪ M2 U M' U2 M U M2 ▪ (y2) M2 U M U2 M' U M2 ▪ R2 U' R' U' R U R U R U' R ▪ (y2) L2 U' L' U' L U L U L U' L ▪ B2 U' M' U2 M U' B2

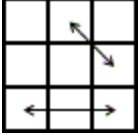
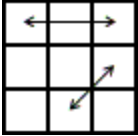
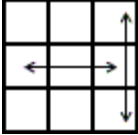
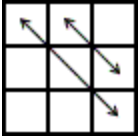
A Permutation : a 	▪ $R' F R' B^2 R F' R' B^2 R^2$ ▪ $(x') R' D R' U^2 R D' R' U^2 R^2$ ▪ $(x) R' U R' D^2 R U' R' D^2 R^2$ ▪ $(y' x) L^2 D^2 L' U' L D^2 L' U L'$ ▪ $(y^2) (L' B L') (F^2 L B' L') (F^2 L^2)$ ▪ $(y^2 z) F^2 R U^2 R' U^2 F^2 L' U^2 L U^2$ ▪ $(y z') U^2 R U^2 R' F^2 U^2 L' U^2 L F^2$ ▪ $R' U^2 R^2 U' L' U R' U' L U R' U^2 R$ ▪ $(y) R^2 F^2 (R' B' R) F^2 (R' B R')$
A Permutation : b 	▪ $R B' R F^2 R' B R F^2 R^2$ ▪ $(x) R D' R U^2 R' D R U^2 R^2$ ▪ $(x') R U' R D^2 R' U R D^2 R^2$ ▪ $(y' x) R^2 D^2 R U R' D^2 R U' R$ ▪ $(y^2 x) L U' L D^2 L' U L D^2 L^2$ ▪ $(y z) U^2 L' U^2 L F^2 U^2 R U^2 R' F^2$ ▪ $(z') F^2 L' U^2 L U^2 F^2 R U^2 R' U^2$ ▪ $(y') R' U^2 R U' L' U R U' L U R^2 U^2 R$ ▪ $(z) (U R' D r^2 U') (R U r^2 U') (D')$ ▪ $(z' y^2) (U' r x) (l^2 U L U') (R' U) (L' U' L') (R' U)$
E Permutation 	▪ $R^2 U R' U' y (R U R' U')^2 R U R' y' R U' R^2$ ▪ $x U R' U' L U R U' L' U R U' L U R' U' L'$ ▪ $y x' R U' R' D R U R' D^2 L' U L D L' U' L$ ▪ $x U R' U' L U R U' r^2 U' R U L U' R' U$ ▪ $y x R U' R' D R U R' u^2 R' U R D R' U' R$ ▪ $l' U' L' U R U' l U R' U' L U R U' l' U$ ▪ $(r' R' U') (L D' L' U L) (R U' R' D R) U$ ▪ $r^2 U r^2 D x' (R U' R' U) 3 x D' r^2 U' r^2$ ▪ $x' U' R U L' U' R' U r^2 U R' U' r' F R F'$ ▪ $y' z' R' F R^2 U R' B' R U' R^2 F' R z R B R'$ ▪ $R' U L' D^2 L U' R L' U R' D^2 R U' L$ ▪ $(R B L B' R') y (R L) y (L B' R' B L') y' (R' L')$ ▪ $l' U' (r' F R F') (R U R' U') L (U R U' R') F$ ▪ $(F' R U R' U') L' U R U' (l' U R' U' r) F R$

Permutations of Edges and Corners:

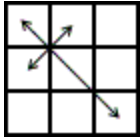
F Permutation	▪ $(y^2) R U' R' U R^2 y R U R' U' x U' R' U R U^2$
----------------------	--

	▪ $yz\ R\ U\ R'\ U'\ R\ U^2\ z'\ y'\ R\ U\ R'\ U'\ y\ x\ L'\ U'\ L\ U\ L^2\ x'$ ▪ $R'\ U\ R\ U'\ R^2\ F'\ U'\ F\ U\ x\ R\ U\ R'\ U'\ R^2\ B'$ ▪ $R'\ U\ R\ U'\ R^2\ F'\ U'\ F\ U\ R\ U'\ x'\ R^2\ U'\ R'\ U$ ▪ $R'\ U\ R\ U'\ R^2\ y'\ R'\ U'\ R\ U\ y\ x\ R\ U\ R'\ U'\ R^2\ x'$ ▪ $y\ R\ U\ R'\ U\ R\ U^2\ R^2\ U'\ R\ U'\ R'\ U^2\ R\ U\ r\ U\ R'\ U'\ L'\ U\ R\ U'$ ▪ $R'\ U^2\ R'\ d'\ R'\ F'\ R^2\ U'\ R'\ U\ R'\ F\ R\ U'\ F$ ▪ $(y)\ U'\ R'\ U\ R\ U'\ R^2\ F'\ U'\ F\ U\ R\ x\ U\ R'\ U'\ R^2$ ▪ $(y^2)\ M\ U^2\ r'\ U\ l'\ U^2\ r\ U'\ R^2\ r\ U^2\ R^2$
G Permutation : a 	▪ $(y)\ R^2\ u\ R'\ U\ R'\ U'\ R\ u'\ R^2\ y'\ R'\ U\ R$ ▪ $(y)\ R^2\ D\ B'\ U\ B'\ U'\ B\ D'\ R^2\ F'\ U\ F$ ▪ $(y^2)\ F^2\ D\ R'\ U\ R'\ U'\ R\ D'\ F^2\ L'\ U\ L$ ▪ $R\ L\ U^2\ R'\ L'\ y'\ R'\ U\ L'\ U^2\ R\ U'\ L$
G Permutation : b 	▪ $R'\ U'\ R\ y\ R^2\ u\ R'\ U\ R\ U'\ R\ u'\ R^2$ ▪ $R'\ U'\ R\ B^2\ D\ L'\ U\ L\ U'\ L\ D'\ B^2$ ▪ $(y^2)\ L'\ U'\ L\ y'\ R^2\ u\ R'\ U\ R\ U'\ R\ u'\ R^2$ ▪ $(y')\ R'\ U\ L'\ U^2\ R\ U'\ L\ y\ R\ L\ U^2\ L'\ R$
G Permutation : c 	▪ $(y)\ R^2\ u'\ R\ U'\ R\ U\ R'\ u\ R^2\ y\ R\ U'\ R'$ ▪ $(y)\ R^2\ u'\ R\ U'\ R\ U\ R'\ u\ R^2\ B\ U'\ B'$ ▪ $(y)\ R^2\ D'\ F\ U'\ F\ U\ F'\ D\ R^2\ B\ U'\ B'$ ▪ $(y^2)\ F^2\ D'\ L\ U'\ L\ U\ L'\ D\ F^2\ R\ U'\ R'$ ▪ $L'\ R'\ U^2\ L\ R\ (y)\ L\ U'\ R\ U^2\ L'\ U\ R'$ ▪ $(y')\ l'\ U'\ U'\ L'\ U\ l\ F'\ U'\ L\ U\ F\ R'\ F\ R$
G Permutation : d 	▪ $(y^2)\ R\ U\ R'\ y'\ R^2\ u'\ R\ U'\ R'\ U\ R'\ u\ R^2$ ▪ $(y^2)\ R\ U\ R'\ F^2\ D'\ L\ U'\ L'\ U\ L'\ D\ F^2$ ▪ $(y^2)\ l^2\ U'\ L^2\ U'\ F^2\ L'\ U'\ R\ U^2\ L'\ U\ l\ x$ ▪ $(y)\ L\ U'\ R\ U^2\ L'\ U\ R'\ y\ R'\ L'\ U^2\ R\ L$ ▪ $(y)\ L\ U'\ R\ U^2\ L'\ U\ R'\ y'\ L'\ R'\ U^2\ L\ R$ ▪ $(y')\ R'\ F'\ R\ F'\ U'\ L'\ U\ F\ R'\ F'\ L\ F^2\ R$
J Permutation : a 	▪ $(y^2)\ R'\ U^2\ R\ U\ R'\ z'\ R^2\ U\ R'\ D\ R\ U'$ ▪ $(y^2)\ L'\ R'\ U^2\ R\ U\ R'\ U^2\ L\ U'\ R\ U$ ▪ $(y^2)\ F^2\ L'\ U'\ r\ U^2\ l'\ U\ R'\ U'\ R^2$ ▪ $(y^2)\ R'\ U^2\ R\ U\ R'\ U^2\ L\ U'\ R\ U\ L'$ ▪ $(y^2)\ x\ U^2\ r'\ U'\ r\ U^2\ l'\ U\ R'\ U'\ R^2$ ▪ $B^2\ R^2\ U'\ R^2\ u\ R^2\ D'\ R^2\ U\ R^2$

	▪ $R U' L' U R' U^2 L U' L' U^2 L$ ▪ $R L d^2 R' U' R d^2 R' U L'$ ▪ $(L' U R' U^2 L U' R) U (L' U R' U^2 L U' R)$ ▪ $L' U R' U^2 L U' R U L' U R' U^2 L U' R$ ▪ $L' U^2 L U L' U^2 R U' L U R'$ ▪ $L' U' L F L' U' L U L F' L^2 U L U$
J Permutation : b 	▪ $R U^2 R' U' R U^2 L' U R' U' L$ ▪ $R U^2 R' U' R d^2 R' U L' U' R$ ▪ $y^2 R L U^2 L' U' L U^2 R' U L' U'$ ▪ $(R U R' F') (R U R' U' R') F (R^2 U' R' U')$ ▪ $y' L' U' R U^2 L' U L U^2 L' R U$ ▪ $y^2 F^2 R U R' y r^2 y' R U' R U R^2$ ▪ $L' U R U' L U^2 R' U R U^2 R'$ ▪ $y^2 r^2 U' L' U' r' U^2 R B' R' U^2$ ▪ $(R U' L U^2 R' U L) U' (R U' L U^2 R' U L)$ ▪ $L' U R' U^2 L U' L' U^2 R L$
N Permutation : a 	▪ $R' U R^2 B^2 U R' B^2 R U' B^2 R^2 U' R$ ▪ $z D R' U R^2 D' R (U' D) R' U R^2 D' R U' R$ ▪ $y' L U' R U^2 L' U R' L U' R U^2 L' U R'$ ▪ $R U' R' U' U F U' R' F' R U' R U' U R'$ ▪ $L U' L' U L F U F' L' U' L F' L F L' U L'$ ▪ $R' U L' U^2 R U' L R' U L' U^2 R U' L$ ▪ $R U' L d^2 L' U L R' U' R U^2 r' F' U' y^2$ ▪ $L U' R U^2 L' U R' L U' R U^2 L' U R' U'$ ▪ $y z R' U R' D R^2 U' R U D' R' D R^2 U' R D'$ ▪ $R U' L d^2 L' U L R' U' R U^2 L' U R U'$ ▪ $F' R U R' U' R' F R^2 F U' R' U' R U F' R'$
N Permutation : b 	▪ $R U' R^2 F^2 U' R F^2 R' U F^2 R^2 U R'$ ▪ $z U' R D' R^2 U R' (U' D) R D' R^2 U R' D R'$ ▪ $L' U R' U^2 L U' L' R U R' U^2 L U' R U'$ ▪ $R' U R U' R' F' U' F R U R' F R' F' R U' R$ ▪ $z U' R D' R^2 U R' U' z' R U R' z R^2 U R' z' R U'$ ▪ $L' U R' z R^2 U R' U z' R U R' z R^2 U R' z' R U'$ ▪ $L' U L U' r' U' F' U L F L' U L' U' r U' L$ ▪ $L U' R U^2 L' U R' L U' R U^2 L' U R' U'$
R Permutation : a	▪ $y^2 z U R^2 U' R^2 U F' U' R' U R U F U^2 R$

	▪ $y' R^2 B' R' U' R' U R B R' U^2 R U^2 R'$ ▪ $R U^2 R' U^2 R B' R' U' R U R B R^2 U$ ▪ $R U^2 R' U^2 R B' R' U' R U I U R^2 F$ ▪ $U^2 R' U' R U^2 R U^2 L' R^2 U R U' R L U^2 R'$ ▪ $y' R U R' F' R U^2 R' U^2 R' F R U R U^2 R' U'$ ▪ $R U^2 R' U' R' F' R U^2 R U^2 R' F R U' R' U$ ▪ $R U^2 R D R' U R D' R' U' R' U R U R' U$ ▪ $y^2 L U^2 L' U^2 L F' L' U' L U L F L^2 U$
R Permutation : b 	▪ $R' U^2 R U^2 R' F R U R' U' R' F' R^2 U'$ ▪ $y R^2 B^2 U' R' U' R U R U B^2 R U' R U$ ▪ $R' U^2 R U' y' R' F R B' R' F' R z x' R' U R'$ ▪ $y^2 R' U^2 I R U' R' U I' U^2 R F R U' R' U' R U R' F'$ ▪ $R' U^2 R' D' R U' R' D R U R U' R' U' R U'$ ▪ $(y x) R^2 U I U R U' I' U' I U^2 R' U^2 R$
T Permutation 	▪ $R U R' U' R' F R^2 U' R' U' R U R' F'$ ▪ $y^2 L' U' L U L F' L^2 U L U L' U' L F$ ▪ $F R U' R' U R U R^2 F' R U R U' R'$ ▪ $L^2 U' L^2 D F^2 R^2 U R^2 D' F^2$ ▪ $R^2 U' R^2 D B^2 L^2 U L^2 D' B^2$ ▪ $R^2 u' R^2 U R^2 y R^2 u R^2 U' R^2$ ▪ $R^2 U R^2 U' R^2 U' D R^2 U' R^2 U R^2 D'$ ▪ $(z x^2) U^2 r' U^2 r U^2 x U^2 r U^2 r' U^2$
V Permutation 	▪ $R' U R' U' y x R' U' R^2 x' U' R' U R' F R F$ ▪ $R' U R' U' B' D B' D' B^2 R' B' R B R$ ▪ $R' U R' U' x^2 y' R' U R' U' I R U' R' U R U$ ▪ $y L' U R U' L U L' U R' U' L U^2 R U^2 R'$ ▪ $y^2 R U' L' U R' U' R U' L U R' U^2 L' U^2 L$ ▪ $y' R' U L U' R U R' U L' U' R U^2 L U^2 L'$ ▪ $L U' R' U L' U' L U' R U L' U^2 R' U^2 R$ ▪ $R' U R' d' R' F' R^2 U' R' U R' F R F$ ▪ $R' U R' U' y z' U' R U' R' U^2 y' R' U' R U R$ ▪ $R' U R' U' y R' F' R^2 U' R' U R' F R F$ ▪ $R' U R' U' y R' D R' D' R^2 F' R' F R F$ ▪ $R' U^2 R U^2 L U' R' U L' U L U' R U L'$ ▪ $R' U^2 R U^2 L U' R' U r' F r U' R U r'$ ▪ $R U' L' U R' U' R U' L U R' U^2 L' U^2 L$

Y Permutation



- $FRU'R'U'RUR'F'RUR'U'R'FRF'$
- $R2uR2'UR2D'R'U'RF2'R'UR$
- $FR'FR2U'R'U'RUR'F'RUR'U'F'$
- $R'U'RF2R'URdR2U'R2'U'R2$
- $R2U'R'URU'z'y'L'U'R'U'R'U'LUyz$
- $yF'L'ULUL'U'LFL'U'LU'F'LF$
- $FRU'R'FDR'yxR'U'RzR2yL'd2$
- $z'U2L'U'LU'U'y'L'U'R'U'R'U'LU$
- $FR'F'RURU'R2U'RUIU'R'U$
- $zU2RUR'U'RyRUL'ULUR'U'$
- $FRU'R'U'RdRUR'B'RUR2$
- $R'FRF'y'U'R'UR2UR'U'R'FRF'U'$
- $y2zU2R'U2R'U2RBRB'U2BR'B'$