

Lin Chen's CFOP (Algorithms)

2013-10-22

First Two Layer (F2L)

First Two Layers, or F2L are normally the first two bottom layers of the 3x3x3 cube, or essentially all layers up until the last layer on larger cubes.

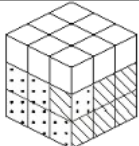
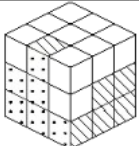
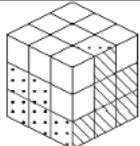
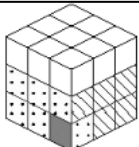
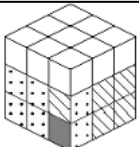
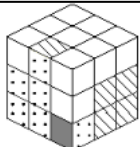
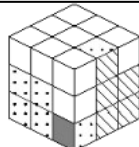
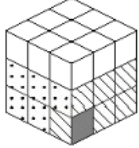
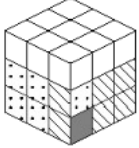
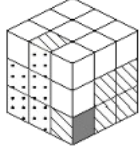
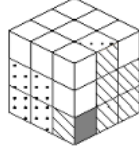
The definition is a little different depending on the subject or who you are talking to. Normally it is as above but it may also refer to the part of the Fridrich method that solves the pairs without counting the cross part.

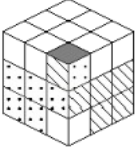
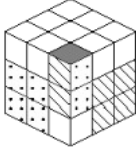
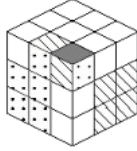
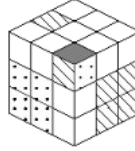
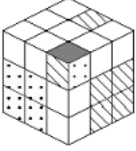
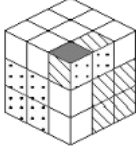
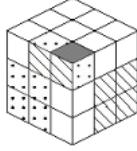
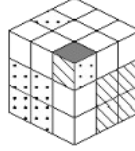
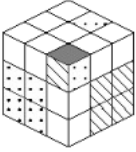
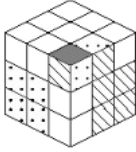
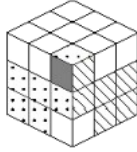
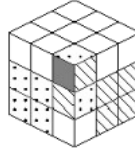
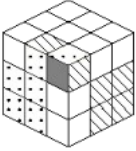
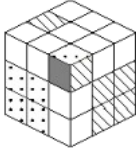
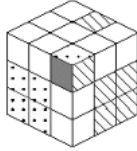
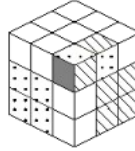
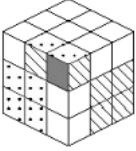
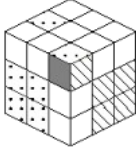
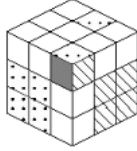
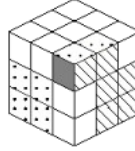
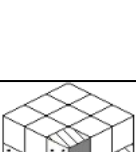
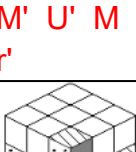
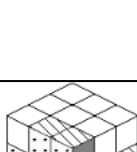
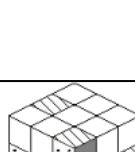
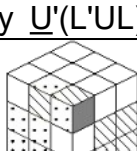
Fridrich F2L

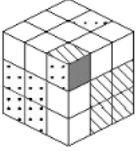
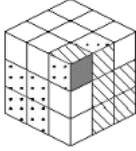
There are many ways to solve the 'F2L' on a cube. A common system is using the Fridrich method first two layer approach. After solving the cross, a corner-edge pair is paired up, and then inserted into the correct slot. A total of four corner edge (or 'CE') pairs are made and inserted to solve the first two layers.

The concept of pairing up four corner/edge pairs was first proposed by René Schoof in 1981.

Algorithms:

 $(R\underline{U}'R'U)y'R'U2)$ $(R'U')(R'UR)$	 $(UR\underline{U}'R'U)y'$ $(R'UR)$	 $y(U'L'UL)(F' r' U$ $r')$ $R'\underline{U}'R'\underline{U}'R'URUR$	
 $(RUR'U')(RU'2R'$ $U')(R'UR')$	 $(RU'R'U)y'(R'U')$ $RU'(R'U'R)$	 $y'(R'\underline{U}'RU)(R'\underline{U}'R)$	 $(RU'R'U)(RU'R')$
 $(RU'R'U)(RU'2R'$ $U)(RU'R')$	 $(RUR'U')(RU'R'$ $U2)y'(R'\underline{U}'R)$	 $y(L'UL\underline{U}')(L'UL)$	 $(RUR'\underline{U}')(RUR')$

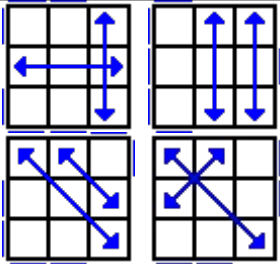
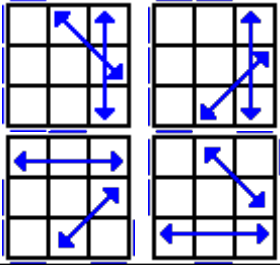
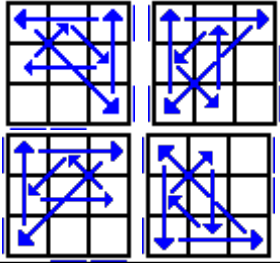
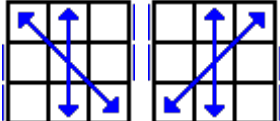
 (RUR'<u>U</u>')(RUR'<u>U</u>) (RUR')	 (R<u>U</u>'R')y'(R'U2 R) <u>U'</u>(R'FRF')(R<u>U</u>'R')	 y'(R'U'2RU) (R'<u>U</u>'R)	 y'<u>U</u>'(R'U'2) (R<u>U</u>'R'U)R
 y'(R'URU'2)R'<u>U</u>'R <u>F'</u> L'U2L F	 y'(R'U'RU'2) (R'<u>U</u>'RU)(R'<u>U</u>'R)	 (RUR'U'2) (RUR'<u>U</u>')(RUR')	 (RU'R'U2)(RUR')
 U(RU'2R'U)RU'R'	 (RU'2R'<u>U</u>')(RUR')	 <u>U</u>'(R<u>U</u>'R'U2) (R<u>U</u>'R')	 <u>U</u>'(RUR')d(R'<u>U</u>'R)
 y'U(R'URU') (R'<u>U</u>'R)	 y' (R'<u>U</u>'R)	 y'U(R'U'RU') (R'U'R)	 U'(RU'R'U2)y' (R'U'R)
 y'(R'U)(RU'2) y (RUR')	 U'(RU'2)(R'U2)(R U'R') <u>M'</u> U' M U2 r U' r'	 U'(RUR'U'2) (R'U'R')	 U (RU'R')
 <u>U</u>(RUR'U'2)RUR'	 y'U(R'FR'F)RU'R	 y <u>U</u>'(L'UL)	 U'(r U'R'U)(RUR')
 y M' U M U'2 R'FR	 (RU'R'U2)y' (R'<u>U</u>'R)	 (RU'R'U)(RU'R') U'2(RU'R')	 <u>U</u>'(RUR'U)(RUR')

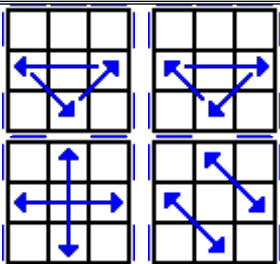
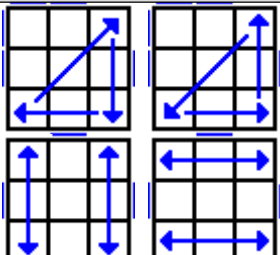
 (RUR')	 $\underline{U}'(\underline{RU}'R'U)(RUR')$		
---	---	--	--

Permutation of the Last Layer

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. The diagrams below are top views of where you want the pieces to go. For example, the T Permutation (or 'T perm') swaps the UL and UR edges, as well as the UFR and UBR corners. Make sure to try out all of the available algorithms for a case to see which one feels the fastest to you - the same algorithm may not be the fastest for everyone, and shorter algorithms are not always faster than longer ones.

Algorithms:

PLL 8-12		08. (RUR'<u>U</u>')(R'F)(R2<u>U</u>'R'<u>U</u>')(RUR'E') 09. (R'<u>U</u>'F'RUR'<u>U</u>')(R'FR2<u>U</u>'R'<u>U</u>'RUR'<u>U</u>R) 10. (R'UR'<u>U</u>') y (R'F'R2<u>U</u>') (R'UR'F)RF 11. F(RU'R'<u>U</u>')(RUR'E')(RUR'<u>U</u>')(I'UR<u>U</u>')
PLL 13-15		12. (R'U'2RU)(R'U'2)z(UR'DRU') 13. (RUR'E'RUR'<u>U</u>')(R'FR2<u>U</u>'R') 14. (R'U'2)(RU'2)(R'FRUR'<u>U</u>')(R'E'R2<u>U</u>') 15. <u>U</u>'(RU'R'<u>U</u>')(RURD)(R'<u>U</u>'RD')(R'U'2R')
PLL 16-19		16. (R'2F2)(RU2')*2(R'FRUR'<u>U</u>')(R'FR2) 17. (RUR')y'(R2 <u>u</u>' RU'R'<u>U</u>')(R' <u>u</u> R2) 18. (R2 <u>u</u>)(R'UR'<u>U</u>')(R<u>u</u>'R'2)(F'dR) 19. (R'd'F)(R2 <u>u</u>)(R' <u>U</u> R <u>U</u> R <u>u</u>' R2)
PLL 20、 21		20. z (UR'DR'2<u>U</u>'RD')*2 21. (R'UR<u>U</u>')(R'F'<u>U</u>'F')(RU'UR'<u>U</u>')(I <u>U</u>'R)

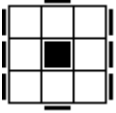
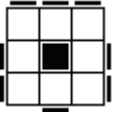
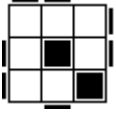
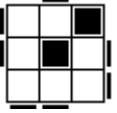
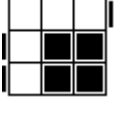
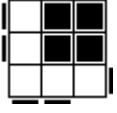
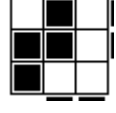
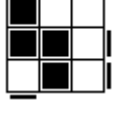
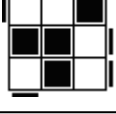
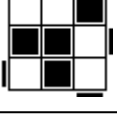
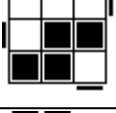
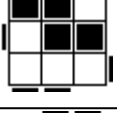
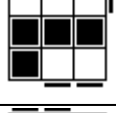
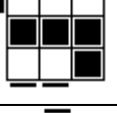
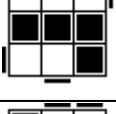
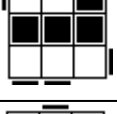
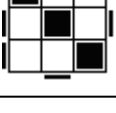
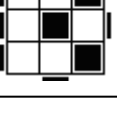
ELL		01.M'2 U M U2 M' U M'2 Or RU'RURURU'R'U'R2 02.M'2 U' M U2 M' U' M'2 Or R2URUR'U'R'U'R'UR' 03.M'2 U M'2 U2 M'2 U M'2 04.M'2 U M' U2 M'2 U2 M' U' M'2
CLL		05.x' R2 DD (R' <u>U</u>' R) DD (R' U R') 06.x' (R <u>U</u>' R) DD (R' U R)DD R'2 07. x'(RU'R'D)(RUR'D')(RUR'D')(RU'R'D') 07' (R2UR'U')y(RUR'U')*2(RUR')y'(RU'R2)

Orientation of the Last Layer

OLL (short for Orientation of the Last Layer) is a last-layer step for 3x3 that orients all last-layer corners and edges in one step. It is the first last-layer step in many speedsolving methods, including the Fridrich Method. OLL is usually followed by PLL.

There are 57 kinds of algorithm for OLL. But you might be skipped this step if you are lucky enough

Algorithms:

	01.(RU'U')(R2'FRF')U2(R'FRF')		02.(r U r')U'2 (RU'2)(R'U'2)(r U' r')
	03.f(R U R' U')f' U' F(R U R' U')F'		04.f(R U R' U')y x(R' F)(R U R' U')F'
	05.(r' U2) (RUR'U) r		06.(r U'2)(R'U'RU'r')
	07.(r U R' U)(R U'2 r')		08.(r' U' R U')(R' U2 r)
	09.(r' R2U'2) R'U'(RU'R'U') M'		10.(RUR'U)(R'FRF')(RU'2R')
	11.r'(R2UR'U)(RU'2R'U)(r R')		12.(r R'2U'RU')(R'U'2RU'R)r'
	13.(r U' r' U')(r U r') (F' U F)		14.R'F(RUR'F')R(F U' F')
	15.(r' U' r)(R'U'RU)(r' U r)		16.(r U r')(RUR'U')(r U' r')
	17.RUR'U (R'FRF')U2(R'FRF')		18.(r U' r' F)(UFU')(RUR'U')F'

	19.(MU)(RUR'U' r)(R'2 FRF')		20.(MU)(RUR'U'r2)(R2'U)(RU') r'
	21.(RU'U')(R'U'RUR'U')(RU'R')		22.RU'U' (R'2U')(R2U')R'2 U'2R
	23.F (RU'R'U)(RUR'U)(RU'R'F')		24.(rUR'U') (r'FRF')
	25.F'(r U R' U') (r' F R)		26.R U' U' R' U' R U' R'
	27.R' U2 R U R' U R		28.M' U' M U2 M' U' M
	29.(r U R'U')(R r'2 FRF')(r R')		30.FU(RU'2R'U')(RU'2R'U')F'
	31.(r' F'UF)(r U' r' U' r)		32.S(RUR'U')(R'FR f')
	33.(RUR'U') (R'FRF')		34.(R'U'R U) y(r U R' U')r' R
	35.(RU'2)(R2'FRF')(RU'2R')		36.(R'U'RU')(R'URU) I U'R'U
	37.F (RU'R'U')(RUR' F')		38.(RUR'U) (R U'R'U')(R'F R F')
	39.(r U' r' U' r) y(RUR' f')		40.(R'F)(RUR'U')(F'UR)
	41.f (RUR'U')f' U' RUR'URU'2R'		42.(R'U'RU')(R'U'2R)(FRUR'U')F'
	43.f' (L' U' L U) f		44.f (R U R' U')f'
	45.F (R U R' U') F'		46.(R'U')(R'FRF')(U R)

	47. B'(R'U'RU)(R'U'RU) B		48. F(RUR'U')(RUR'U') F'
	49. r U' r'2 U r2 U r'2 U' r		50. (R'URU'R'2 F R2) (UR'U'F'R)
	51. f (RUR'U')(RUR'U') f'		52. R'U'RU'R'U y r' F r F
	53. (r' U2)(RUR'U') (RUR'U) r		54. (r U'2)(R'U'RUR'U')(R U' r')
	55. R'F(URU'R'2F'R2)(UR'U'R)		56. (r' U' r)(U'R'UR)(U'R'UR)(r' U r)
	57. (RUR'U' r)(R' U) (R U' r')		58. OLL Skip

The meaning of Alphabet

R=Right
L=Left
U=Up
D=Down
F=Front
B=Back

