01  (R U' R') Dw (R' U2) (R U2') (R' U R)

02  [U] (R U' R') Dw' (L' U L)

03  [Dw'] (L' U L d) (R U' R')

04  (R U R' U') (R U R' U') (R U R')

05  [Dw] (R' U) (R U2) (R' U R)

06  [U'] (R U') (R' U2) (R U' R')

07  (R2' U2) (R' U' R U') (R' U2 R')

08  (R U' R U) y (R U') (R' F2)

09  [U2] (R U' R') y (L' U' L)

10  (R U') (R' U) (R U' R')

11  (R U2') (R U R' U) (R U2' R2')

12  (R U' R') (F' L' U2) (L F)

13  (R U2 R') y' (R' U2 R)

14  (R U R' U') (R U R')

15  (R U' R' Dw) (R' U R)

16  y' (R' U2 R U) (R' U' R)

17  [U2] (R2 U2') (R' U' R U' R2')

18  [Dw'] (L' U2 L U') (L' U L)

19  [U2] (R U R' U) (R U' R')

20  (R U2 R' U') (R U R')

21  y' [U2] (R2' U2) (R U R' U R2)

22  [U] (R U2' R' U) (R U' R')

23  (F' L' U2) (L F)

24  [Dw] (R' U' R Dw') (R U R')

25  [Dw'] (L' U L)

26  [Dw] (R' U2 R Dw') (R U R')

27  [Dw] (R' U' R U2') (R' U R)

28  [Dw] (R' U R Dw') (R U R')

29  [U'] (R U' R' U) (R U R')

30  (R U') (R' U2) y' (R' U' R)

31  (R U R')

32  [Dw] (R' U2 R U2') (R' U R)

33  [U'] (R U R' Dw) (R' U' R)

34  [Dw] (R' U R U') (R' U' R)

35  (Lw U L F') (L' U' Lw')

36  y' (R' U' R)

37  [U'] (R U2' R' U2) (R U' R')

38  [U] (R U' R')

39  [U'] (R U2' R' Dw) (R' U' R)

40  [U'] (R U R' U2) (R U' R')

41 [U'] (R U' R' Dw) (R' U' R)

00

This is Bob Burton's Printable F2L Page
http://www.cubewhiz.com
Questions or comments?
E-mail me at bob@cubewhiz.com

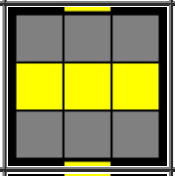
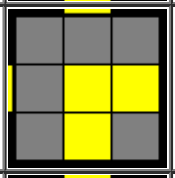
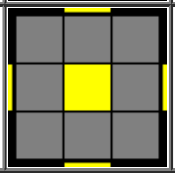
2 Look OLL

These are the 10 orientation cases for orienting the last layer in only two looks. These algorithms appear EXACTLY as I perform them when I am solving the last layer, in speedcubing notation with rotations included in the algorithm. It should be noted that these are the algorithms that I find easiest to perform. However, you may find other algorithms better-suited for your own hands, so it is recommended to try many different algorithms for the same situation to find which one works best for your own style of cubing.

In each diagram, yellow is the color of the upper face. A yellow "bar" indicates that the last layer color is facing that direction in that location. Grey denotes that a particular piece is not oriented correctly.

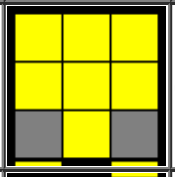
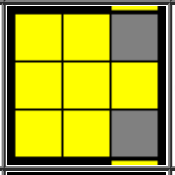
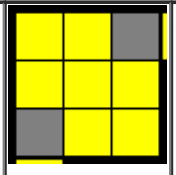
Orient Edges

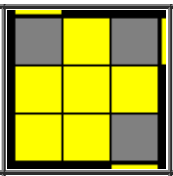
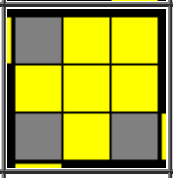
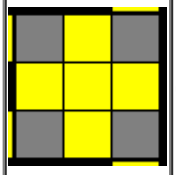
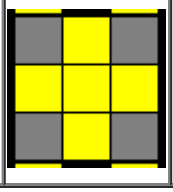
You should already know these three sequences from my beginner method.

Name	Diagram	Algorithm	Comments
Bar		**F (R U R' U') F'**	This is just R U R' U' with a F setup move.
L		**Fw (R U R' U') Fw'**	This is the same sequence as above except with a double F and F' turn.
All		**F (R U R' U') F' Fw (R U R' U') Fw'**	This is just "Edges L" followed by "Edges Bar."

Orient Corners

These are the algorithms that orient corners.

#	Diagram	Algorithm	Comments
23		**(R2' D) (R' U2) (R D') (R' U2 R')**	This is one of the most awkward OLLs with all correctly flipped edges. It's still pretty fast, though.
24		**(Lw' U') (L U) (R U') (Rw' F)**	This one is pretty fast. The only problem is that you have to alternate hands a couple times. I recommend being able to do the mirror of this case as well.
25		**(R' F) (R B') (R' F') (R B)**	This algorithm is ridiculously fast. Hold your thumb on the bottom of the cube and left middle finger holding the S layer on the upper face. It helps to use your wrist of your left hand to assist in some of the turns.

27		**(R U R' U) (R U2 R')**	This is the Sune. Your right hand should never come off of the cube during the execution at any time.
26		**(R U2) (R' U' R U' R')**	This is just the inverse of the Sune, called the Antisune.
22		**(R U2') (R2' U') (R2 U') (R2' U2' R)**	The execution of this algorithm is pretty neat. The R2 turns should alternate in direction so that they can be performed by the right hand without letting go of the cube. The left hand holds the cube and makes the U' turns.
21		**(R U2) (R' U' R U R' U' R U' R')**	This is a very easy case. This can be performed as a double Sune or double Antisune. I recommend learning the COLL cases for this one as well since there are only four cases.

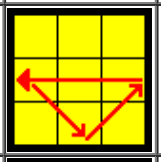
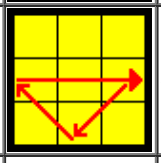
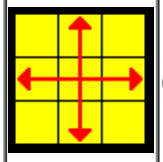
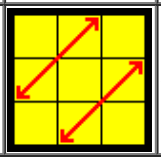
2 Look PLL

These are the 7 permutation cases for permuting the last layer in only two looks. These algorithms appear EXACTLY as I perform them when I am solving the last layer, in speedcubing notation with rotations included in the algorithm. It should be noted that these are the algorithms that I find easiest to perform. However, you may find other algorithms better-suited for your own hands, so it is recommended to try many different algorithms for the same situation to find which one works best for your own style of cubing.

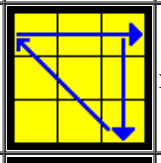
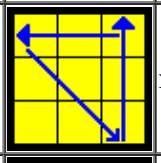
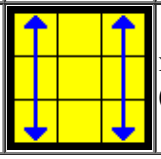
In each diagram, the edges that are being swapped or moved are denoted by the red arrows, while the corners that are being swapped are moved are shown with blue arrows.

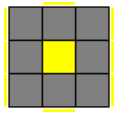
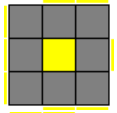
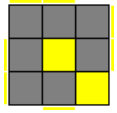
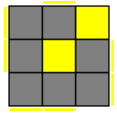
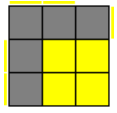
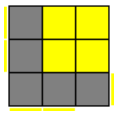
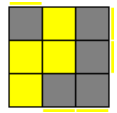
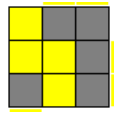
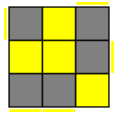
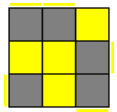
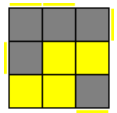
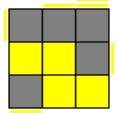
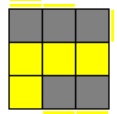
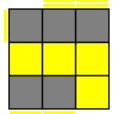
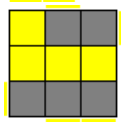
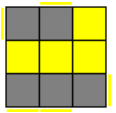
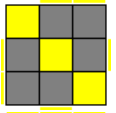
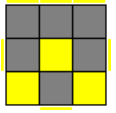
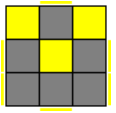
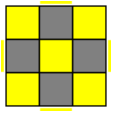
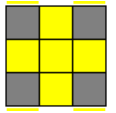
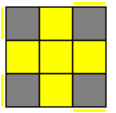
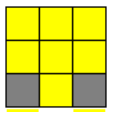
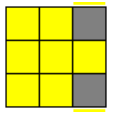
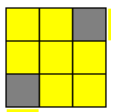
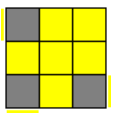
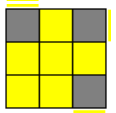
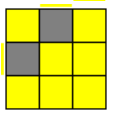
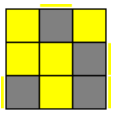
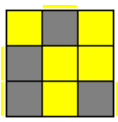
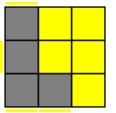
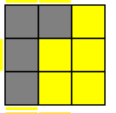
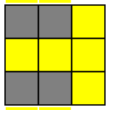
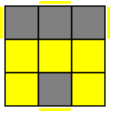
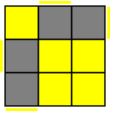
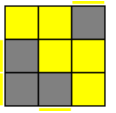
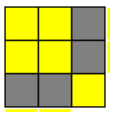
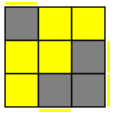
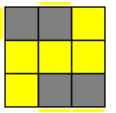
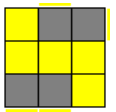
If you'd like to know how I recognize PLLs, check out my PLL Recognition page. For a printable page of these algorithms, visit my printable page. Please note that you will need Adobe Reader to access and print the printable page.

Permute Edges

Name	Diagram	Algorithm	Comments
Ua		**(R U' R U) (R U) (R U') (R' U' R2)**	This is just a simple 3-edge cycle. It is almost as fast as the corner cycles. I solve this case with the bar at the front or the back.
Ub		**(R2 U) (R U R' U') (R' U') (R' U R')**	This is the inverse of the other U perm. I place my hands slightly differently for this algorithm. I solve this case with the bar at the front or the back.
H		**(M2' U) (M2' U2) (M2' U) M2'**	This is extremely easy to recognize and can be performed VERY quickly. The M'2 is actually performed as (M'M') with rapid pushing at the back face of the M layer with the ring and then middle fingers.
Z		**(M2' U) (M2' U) (M' U2) (M2' U2) (M' U2)**	The Z permutation is performed very similarly to the H perm. The last U2 is not necessary if you account for it before the algorithm.

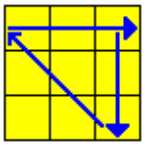
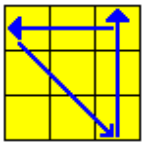
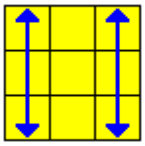
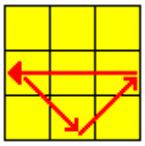
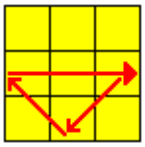
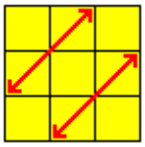
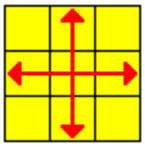
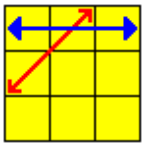
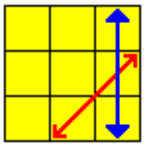
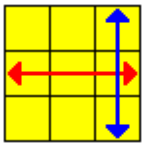
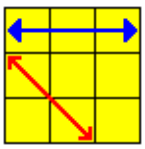
Permute Corners

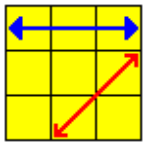
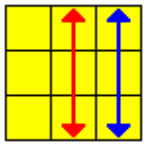
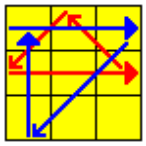
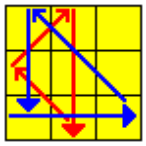
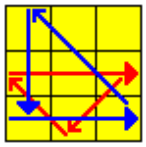
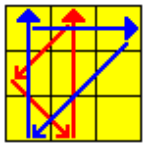
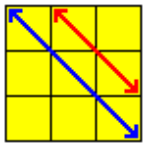
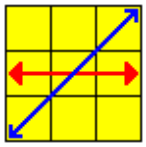
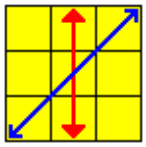
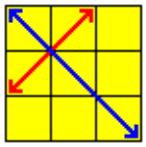
Name	Diagram	Algorithm	Comments
Aa		**x (R' U R') D2 (R U' R') D2 R2**	This is a basic corner 3-cycle. It is one of my favorite and fastest algorithms. Perform the D2s with the left hand and everything else with the right.
Ab		**x R2 D2 (R U R') D2 (R U' R)**	This is just the inverse of the other A perm. It is performed in a very similar manner.
E		**x' (R U') (R' D) (R U R' D') (R U R' D) (R U') (R' D')**	This alg is just two orientations performed consecutively.

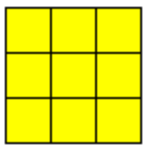
#	Diagram	Algorithm
01		(R U2) (R2' F R F') U2' (R' F R F')
02		F (R U R' U') S (R U R' U') Fw'
03		Fw (R U R' U') Fw' U' F (R U R' U') F'
04		Fw (R U R' U') Fw' U F (R U R' U') F'
05		(Rw' U2) (R U R' U Rw)
06		(Rw U2) (R' U' R U' Rw')
07		(Rw U R' U) (R U2 Rw')
08		(Rw' U' R U') (R' U2 Rw)
09		(R U R' U' R' F) (R2 U R' U' F')
10		(R U R' U) (R' F R F') (R U2 R')
11		Rw' (R2 U R' U R U2 R') U M'
12		(M U2) (R' U' R U') (R' U2 R) U M'
13		(Rw U' Rw' U' Rw U Rw' y' (R' U R)
14		(R' F) (R U R' F' R) y' (R U' R')
15		(Lw' U' Lw) (L' U' L U) (Lw' U Lw)
16		(Rw U Rw') (R U R' U') (Rw U' Rw')
17		(R U R' U) (R' F R F') U2 (R' F R F')
18		F (R U R' U) y' (R' U2) (R' F R F')
19		Rw' (R U) (R U R' U' Rw) x (R2' U) (R U')
20		Rw' (R U) (R U R' U' Rw2) (R2' U) (R U') Rw'
21		(R U2) (R' U' R U R' U' R U' R')
22		(R U2') (R2' U') (R2 U') (R2' U2' R)
23		(R2' D) (R' U2) (R D') (R' U2 R')
24		(Lw' U') (L U) (R U') (Rw' F)
25		(R' F) (R B') (R' F') (R B)
26		(R U2) (R' U' R U' R')
27		(R U R' U) (R U2 R')
28		(M' U M) U2 (M' U M)
29		(L2 U' L B) (L' U) (L2 U') (Rw' U' Rw)
30		(R2' U R' B') (R U') (R2' U) (Lw U Lw')
31		(R' U') F (U R U' R') F R
32		(R Dw) (L' Dw') (R' U) (Lw U Lw')
33		(R U R' U') (R' F R F')
34		(R U R2' U') (R' F) (R U) (R U') F'
35		(R U2) (R2 F) (R F' R U2 R')
36		(L' U' L U') (L' U L U) (L F' L' F)
37		F (R U') (R' U' R U) (R' F')
38		(R U R' U) (R U' R' U') (R' F R F')
39		(L F') (L' U' L U) F U' L'
40		(R' F) (R U R' U') F' U R
41		(R U') (R' U2) (R U) y (R U') (R' U' F')
42		(L' U) (L U2') (L' U') y' (L' U) (L U F)
43		Fw' (L' U' L U) Fw
44		Fw (R U R' U') Fw'
45		F (R U R' U') F'
46		(R' U') (R' F R F') (U R)
47		F' (L' U' L U) (L' U' L U) F
48		F (R U R' U') (R U R' U') F'
49		(R' F R' F' R2) U2 y (R' F R F')
50		(R B' R B R2') U2 (F R' F' R)
51		Fw (R U R' U') (R U R' U') Fw'
52		(R U R' U R Dw') (R U' R' F')
53		(Rw' U') (R U') (R' U) (R U') (R' U2 Rw)
54		(Rw U) (R' U) (R U') (R' U) (R U2' Rw')
55		(R U2) (R2 U' R U' R' U2) (F R F')
56		Fw (R U R' U') Fw' F (R U R' U') (R U R' U') F'
57		(R U R' U') (M' U R U') Rw'

This is Bob Burton's Printable OLL Page
http://www.cubewhiz.com

Questions or comments?
E-mail me at
bob@cubewhiz.com

#	Diagram	Algorithm
Aa		x (R' U R') D2 (R U' R') D2 R2
Ab		x R2 D2 (R U R') D2 (R U' R)
E		x' R U' R' D R U R' D' R U R' D R U' R' D'
Ua		(R U' R U) (R U) (R U') (R' U' R2)
Ub		(R2 U) (R U R' U') (R' U') (R' U R')
Z		(M2' U) (M2' U) (M' U2) (M2' U2) (M' U2)
H		(M2' U) (M2' U2) (M2' U) M2'
Ja		(R' U L') U2 (R U' R') U2 ([L R] U')
Jb		(R U R' F') (R U R' U') (R' F) (R2 U') (R' U')
T		(R U R' U') (R' F) (R2 U') (R' U' R U) (R' F')
Ra		(L U2') (L' U2') (L F' L' U' L U) (L F L2' U)

#	Diagram	Algorithm
Rb		(R' U2) (R U2) (R' F R U R' U') (R' F' R2 U')
F		R' U' F' (R U R' U') (R' F) (R2 U') (R' U' R U) (R' U R)
Ga		(R2' Uw) (R' U R' U' R Uw') R2' y' (R' U R)
Gb		(R' U' R) y (R2' Uw R' U) (R U' R Uw' R2')
Gc		(R2' Uw' R U') (R U R' Uw R2) y (R U' R')
Gd		(R U R') y' (R2' Uw' R U') (R' U R' Uw R2)
V		(R' U R' Dw') x (Lw' U R' U') (Lw R U') (R' U R U)
Na		(R U') (R' U) (Lw U) (F U') (R' F') (R U' R U) (Lw' U R')
Nb		(L' U) (L U') (Rw' U') (F' U) (L F) (L' U L' U') (Rw U' L)
Y		(F R U') (R' U' R U) (R' F') (R U R' U') (R' F R F')



This is Bob Burton's Printable Permutations Page
http://www.cubewhiz.com
Questions or comments?
E-mail me at bob@cubewhiz.com