

Yuxuan's 4 Look Last Layer Tutorial

This document attempts to explain how to do 4 look last layer for 3x3. It assumes that you know how to read notation.

Note 1: The pictures will be presented in the angle you are supposed to hold the cube at to perform the algorithm.

Note 2: All algorithms will also be broken down into triggers (a sequence of moves that is fast and easy to execute). Each case will have two cells in the algorithm section. The first cell will have the normal algorithm(s). The second cell will have the same algorithm(s) broken down into triggers.

Note 3: For the most part, all algorithms will be written the way I execute them (some double turns will have a ' even though a 180 degree turn is the same whether the layer was turned clockwise or counter clockwise). This is to help you fingertrick the algorithm.

Note 4: My philosophy for learning algorithms was/is picking algorithms that are easy to learn (even if it might mean it is more moves/slower) so many of my algorithms build on each other.

Note 5: If there are any problems with the algorithms, contact Yuxuan.

This guide assumes that you know how to solve the first two layers (F2L) and begins right after the first two layers are solved. In other words, this guide is for last layer (LL) only.

4LLL is split into 2 look OLL and 2 look PLL

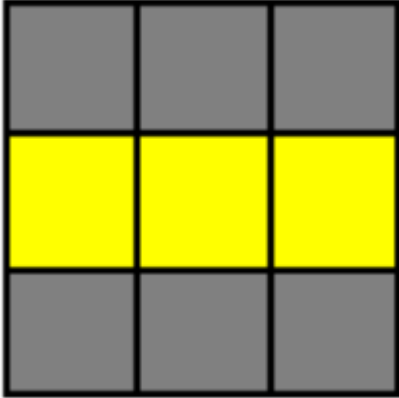
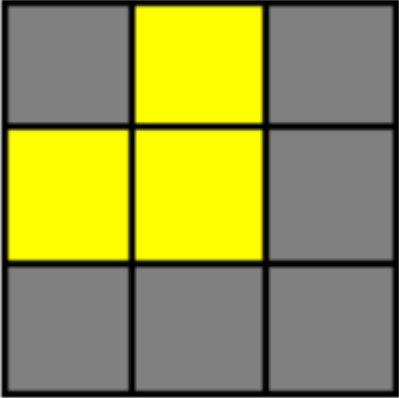
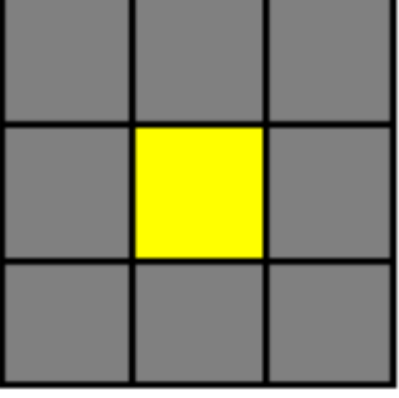
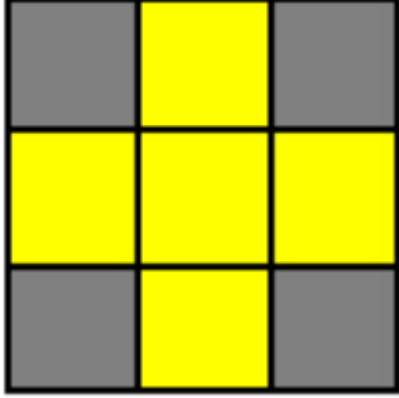
2 Look OLL

Step 1. Orient Edges

(This is the same as the Edge Orientation step in the MCC Beginner Method)

Orient the edges of the last layer by using FRURUF from the correct angle.

There are 4 possible cases (one of them is solved/skip).

<p>Case 1</p>  <p>F R U R' U' F'</p>	<p>Case 2</p>  <p>F R U R' U' R U R' U' F' (Case 1 twice)</p>
<p>Case 3</p>  <p>F R U R' U' F' U2 F R U R' U' R U R' U' F' (Case 1 + Case 2)</p>	<p>Case 4</p>  <p>Solved</p>

Step 2. Orient Corners

There are 8 cases (1 of them is solved/skip)

All edges oriented

OLL	Algorithm(s)	Comments
	$R U R' U R U2' R'$ With triggers: $(R U R' U) (R U2' R')$	"Sune"
	$R' U' R U' R' U2' R$ With triggers: $(R' U' R U') (R' U2' R)$	"Antisune"
	<ol style="list-style-type: none"> $R U2' R' U' R U R' U' R U' R'$ $F (R U R' U')3 F'$ (expanded: $F R U R' U' R U R' U' R U R' U' F'$) With triggers: <ol style="list-style-type: none"> $(R U2' R') U' (R U R' U') (R U' R')$ $F (R U R' U')3 F'$ 	"Double Sune" <ol style="list-style-type: none"> Happens to solve corners if $UBR == UFR$ and $UBL == UFL$ Solves corners if $UBL == UBR$ and $UFL == UFR$
	$R U2' R2' U' R2 U' R2' U2' R$ With triggers: $(R U2') (R2' U') (R2 U') (R2' U2') R$	"Pi" Solves corners if $UBR == UFR$ (and UBL and UFL are opposite colors)
	<ol style="list-style-type: none"> $R2 D' R U2' R' D R U2 R$ $L2' D L' U2 L D' L' U2 L'$ $(U2) R U R' U R U2' R2' U' R U' R' U2' R$ With triggers: <ol style="list-style-type: none"> $R2 D' (R U2' R') D (R U2 R)$ $L2' D (L' U2 L) D' (L' U2 L')$ $(U2) (R U R' U) (R U2' R') (R' U' R U') (R' U2' R)$ 	"Headlights", "U" <ol style="list-style-type: none"> Solves corners if $UBL == FUR$ Solves corners if $UBR == FUL$ Solves corners if UFL and UFR are opposite colors and the back 2 corners are solved. Algorithm is basically sune + antisune
	$r U R' U' r' F R F'$ With triggers: $(r U R' U') (r' F R F')$	"Chameleon" Very similar to the case below
	$F' r U R' U' r' F R$ With triggers: $F' (r U R' U') (r' F R)$	"Triple Sune", "Bowtie" Very similar to the case above (move the last move to the front)

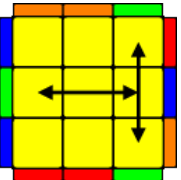
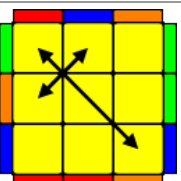
2 Look PLL

Step 1: Solve Corners

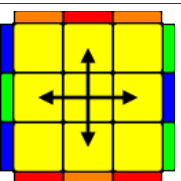
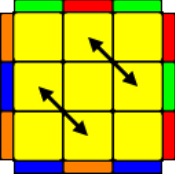
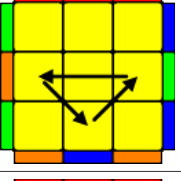
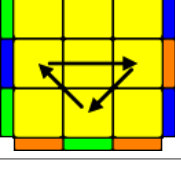
Look at the corners. If they are solved relative to each other, skip to the next step.

If they are not solved relative to each other, see if the corners need an adjacent or diagonal swap.

Corners Only

PLL	Algorithm(s)	Comments
	$R U R' U' R' F R^2 U' R' U' R U R' F'$ With triggers: $(R U R' U') (R' F) (R^2 U' R' U') (R U R' F')$	T perm Look for headlights, place on left
	$F R U' R' U' R U R' F' R U R' U' R' F R F'$ With triggers: $F (R U' R' U') (R U R' F') (R U R' U') (R' F R F')$	Y perm

Step 2: Solve edges

PLL	Algorithm(s)	Comments
	$M^2 U' M^2 U^2 M^2 U' M^2$ With triggers: $(M^2 U' M^2) U^2 (M^2 U' M^2)$	H perm
	$M^2 U' M^2 U' M' U^2 M^2 U^2 M' U^2$ With triggers: $(M^2 U') (M^2 U') (M' U^2) (M^2 U^2) (M' U^2)$	Z perm
	$R U' R U R U R U' R' U' R^2$ With triggers: $(R U' R U) (R U R U') (R' U' R^2)$	Ua perm Not opposite color on left Opposite color on right
	$R^2 U R U R' U' R' U' R' U R'$ With triggers: $(R^2 U) (R U R' U') (R' U' R' U) R'$	Ub perm Opposite color on left Not opposite color on right Inverse of Ua perm