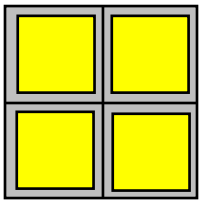
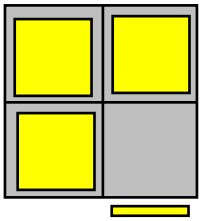
# Orient Second Layer (Bottom Sticker Pattern)

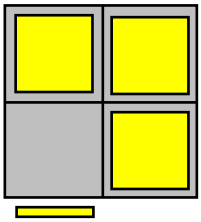
Solved



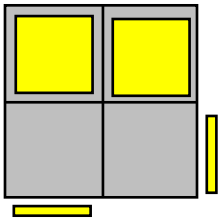
L F U' F'



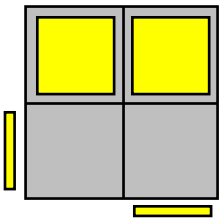
R' F' U F



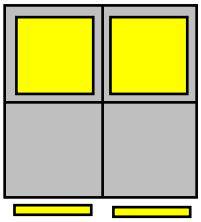
F U' R U R'



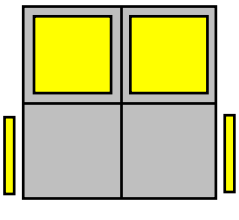
F' U L' U' L



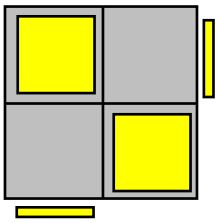
B U2 B2 U' x'



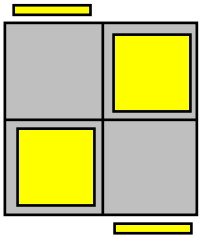
F L2 U' F2



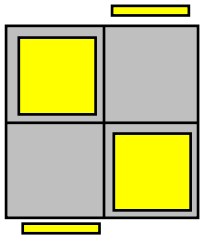
R2 U B2 R'



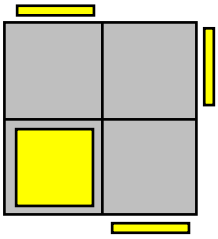
R2 F' R2 U' R2 x



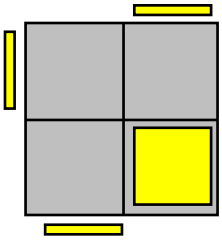
L2 F L2 U L2 x



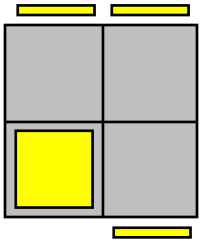
F' L2 U' z



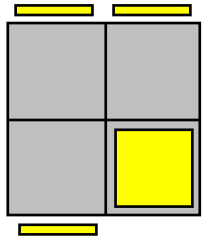
F R2 U z'



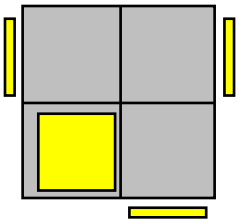
F U F' D2 F' R2 x



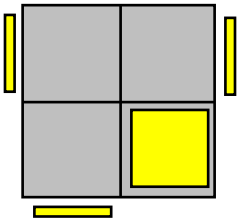
F' U' F D2 F R2 x'



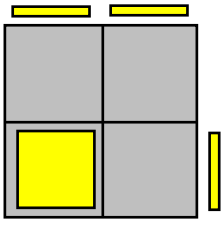
F' U' L' D2 L' F2 z



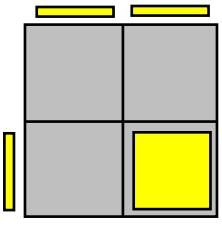
F U R D2 R F2 z



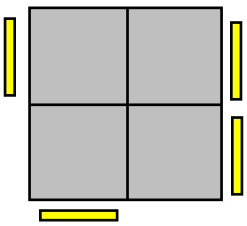
F U F2 U F U'



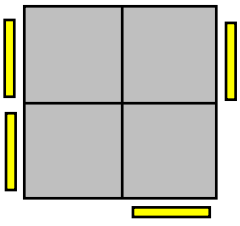
F' U' F2 U' F' U



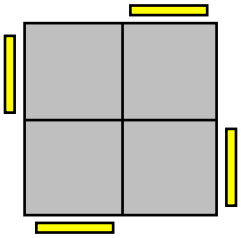
L U L U z



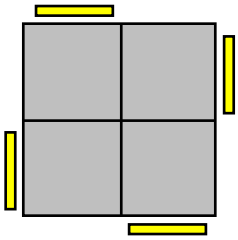
R' U' R' U' z'



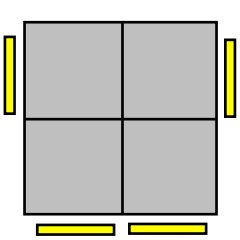
R D R D z



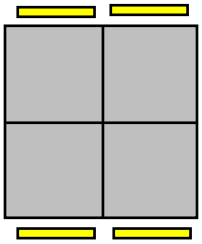
L' D' L' D' z'



F2 D F2 R U2 z'



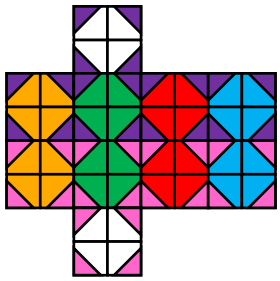
F2 D2 x



# Notes on Notation

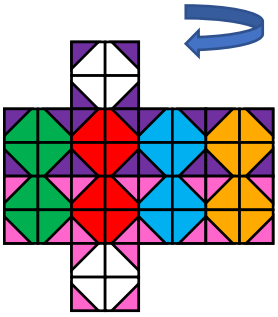
Notation for the puzzle moves I struggled with because of the discrepancy between the number of moves possible on any face at a time. This is the notation I settled on because it made the most sense to me. As well as possible I tried to follow the already existing cube notation for a regular Rubik's cube while adding and subtracting as necessary. This notation only applies to the whole puzzle and the section on second layer Orientation uses a subset of this. I found it was easiest to use the puzzle with it held so that it stood tall in your hands, so that will be the assumed orientation if I ever reference left, right, up, down front or back. For all moves marked as 90° rotations the opposite direction will be marked Prime (').

Assumed beginning orientation and state

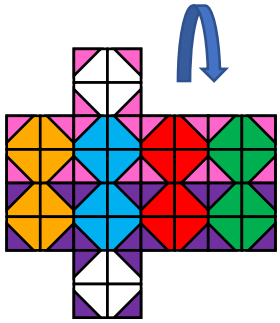


Rotations : (Do not change state of puzzle only moves available)

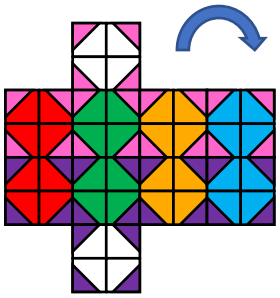
z (90°)



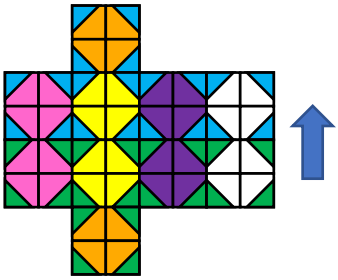
x (180°)



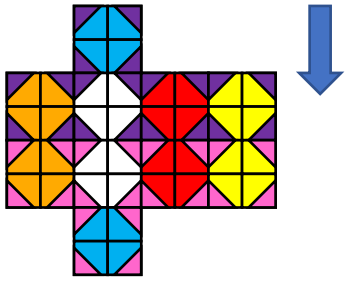
y (180°)



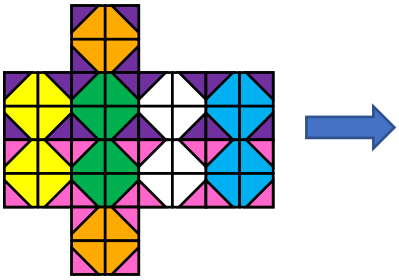
g (the gyro move as outlined by Melida Green)



ux (90°)

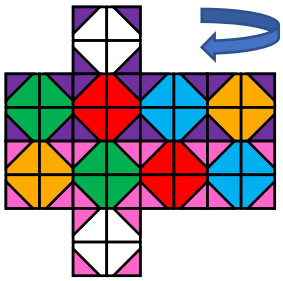


uy (90°)

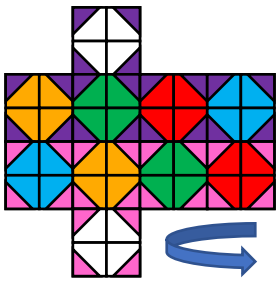


Basic Moves :

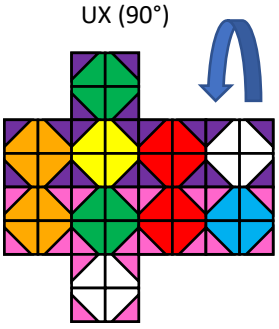
U (specifically UZ) (90°)



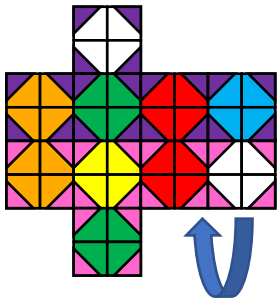
D (specifically DZ) (90°)



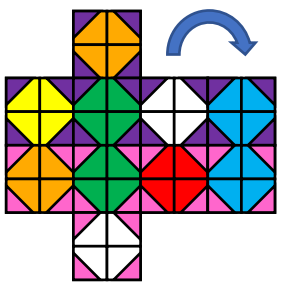
UX (90°)



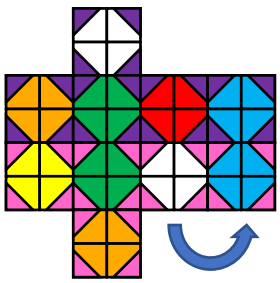
DX (90°)



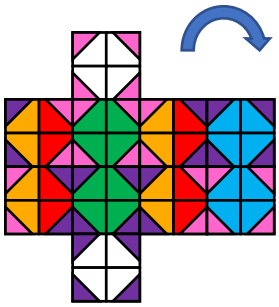
UY (90°)



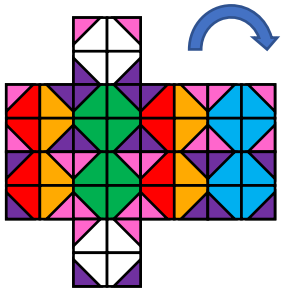
DY (90°)

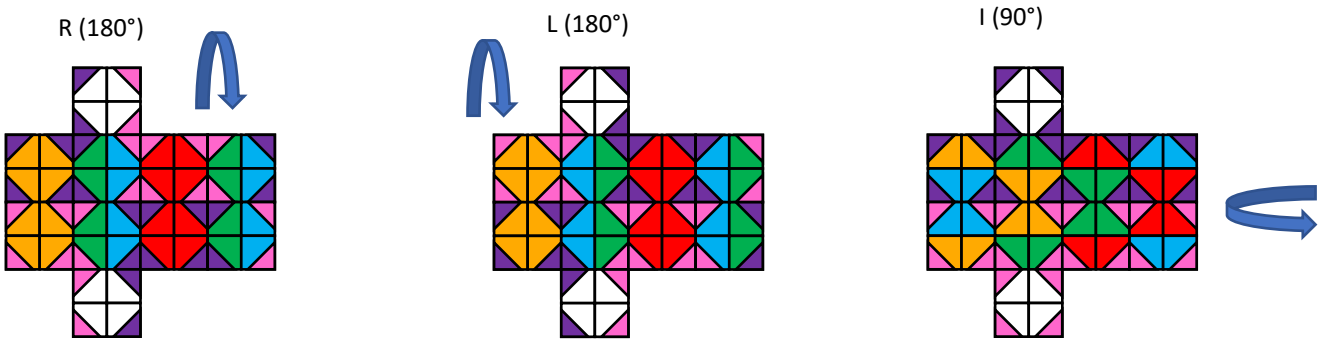


F (180°)



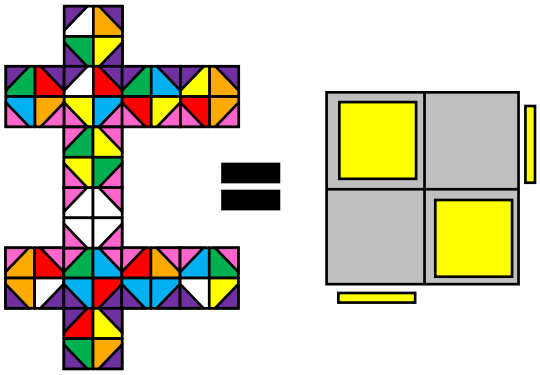
B (180°)






## Notes on Orient Second Layer

When using this method, it is most efficient to solve the orientation of stickers on one of the four layers, just like the first step in Ortega, and then transfer the remaining layer on that half of the puzzle to the other side in order to solve the orientation faster. When doing this it can be harder to keep track of the pieces so these 23 algorithms are ways to solve the stickers fast without thinking. These are not optimized for ergonomics or move count but are the fastest that I could come up with. The algorithms start with the puzzle having the stickers needed oriented occupying the 2nd layer and the other half of the puzzle having the already solved stickers in the 3rd or 4th layer entirely. Once in this state the moves given correspond to the moves needed to be done on the equivalent 2x2 puzzle in the top half of the puzzle.

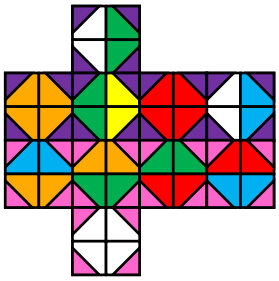


## Special Notation

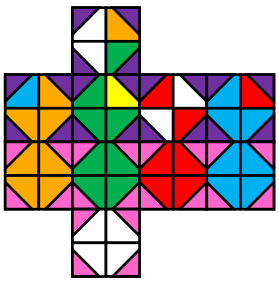
For orientation of the second layer and other parts of this method throughout the method of performing moves on the upper half of the cube as if it were a 2x2x2 is employed. To achieve this, I use the other half of the puzzle as a buffer and combine UX and UY moves with I moves to effectively isolate the top half and turn the sides of it freely. What is nice about this is that any move can be accomplished by placing the correct face down towards the center, like the right side if an r or r' move is wanted and then to move the I face in the same way I for r and I' for r'. Because it is tedious to write out, I employ parentheses to denote when I am talking about moves on just the top half of the puzzle and these can be achieved using this method. This notation is used entirely in the Orient Second Layer Portion

Brackets are different than parentheses and denote a repetition of the enclosed moves a specified number of times

( r ) = UY I UY'

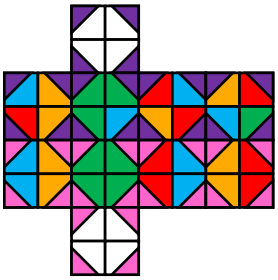


( r u r' u' ) = UY I UY I UY' I' UY I' UY2

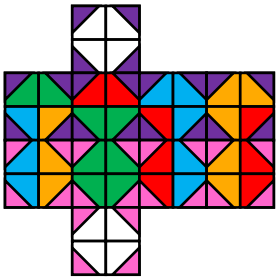


# Permute All Layers (PAL)

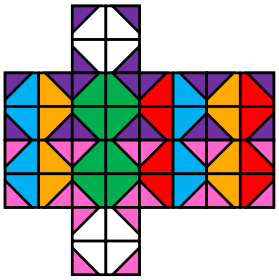
B-X-B-B (16/81)
I R F I' U R U' R I R U' F U F



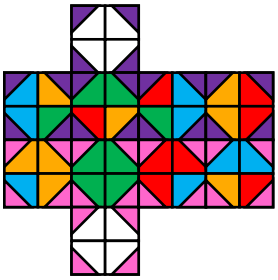
S-B-B-B (16/81)
I R U' F U F I R U F I' U' R



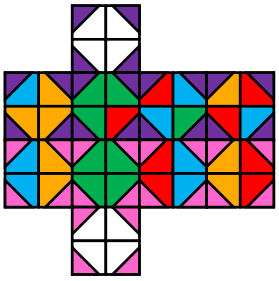
B-B-B-B (4/81)
R U' B U2 R U' R



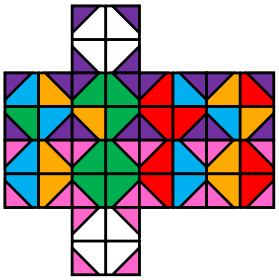
B-X-S-B (8/81)
U' R U2 F U F I U2 F I U R U2 F



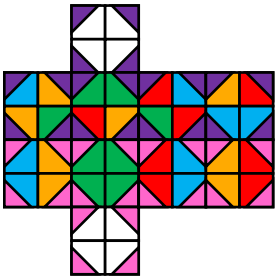
B-L-B-B (4/81)
U2 F U' R I R U' F U F I' U2 R U' F R E



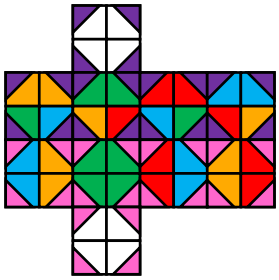
B-R-B-B (4/81)
U2 F U F I' F U F U' R I' U2 R F U' F



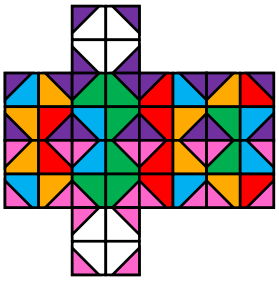
B-C-B-B (4/81)
F U2 R I' R U R U2 F U R I' R U2 F



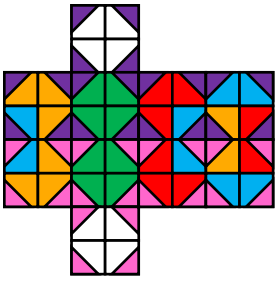
S-X-B-B (4/81)
E2 R U2 F I R E2 U R U R U' F U R



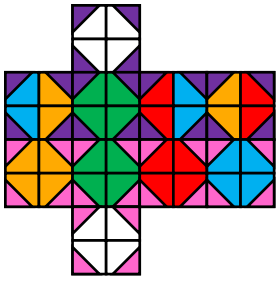
B-X-X-B (4/81)
U F U2 R U R I F U R E2 U2 F



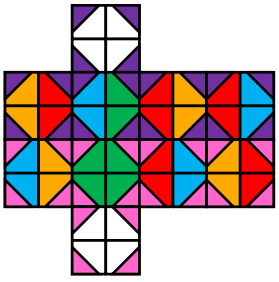
S-B-B-S (4/81)
F U R I R U2 F I F I' U F



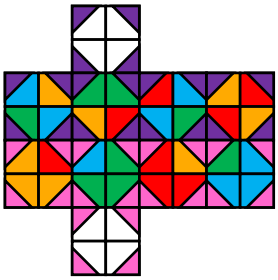
B-B-S-S (2/81)
R U R U' R F U' F U F



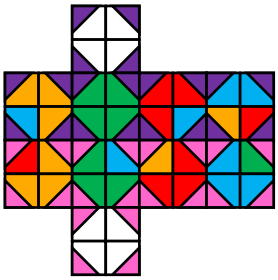
X-X-B-B (2/81)
R U R U' R U R U' R



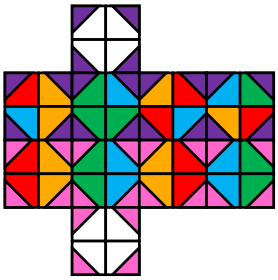
B-X-X-S (2/81)
R U' R F I F U F U' R I F U' F U



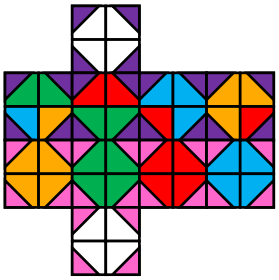
S-B-X-S (2/81)
R U' R I' R U' R U R I R U R



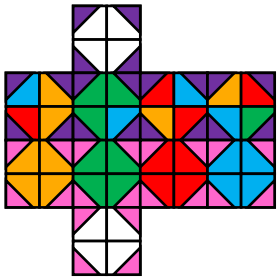
X-B-X-X (1/81)
R U R I' R U R I' R U R I' R U F R E



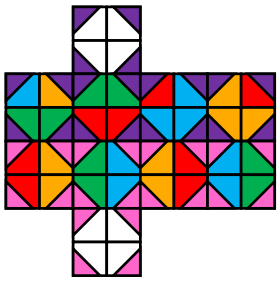
S-B-S-S (1/81)
U2 R U R I' R U R I' R U R I' R U R



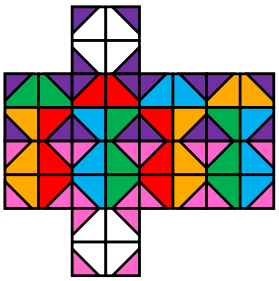
B-X-S-S (1/81)
R U R I' U2 R U' R F U' R I' U2 R U F



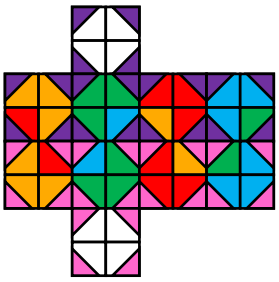
B-S-X-X (1/81)
I R U' F I' U2 F U R U' F I U2 F U R



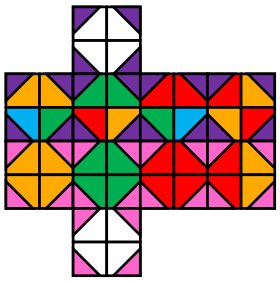
S-X-X-X (1/324)
R U2 F U F I R U2 F I R U' F U2 R



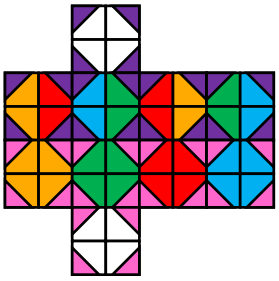
S-X-X-S (1/324)
R U2 F I F E2 U2 R



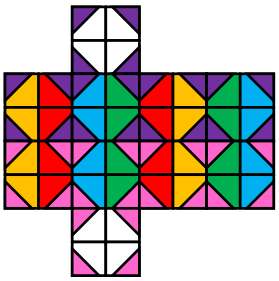
S-X-S-S (1/324)
R U2 F U R I R U F I' F U F U2 R



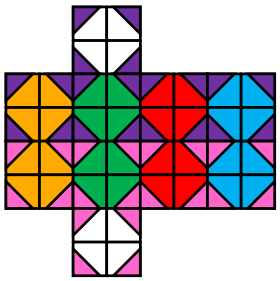
X-X-S-S (1/648)
R U2 F U R U2 F U F U2 R

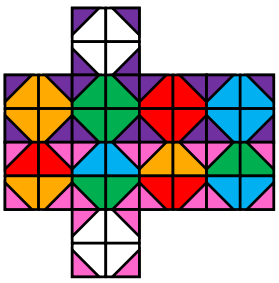


X-X-X-X (1/1296)
R F R



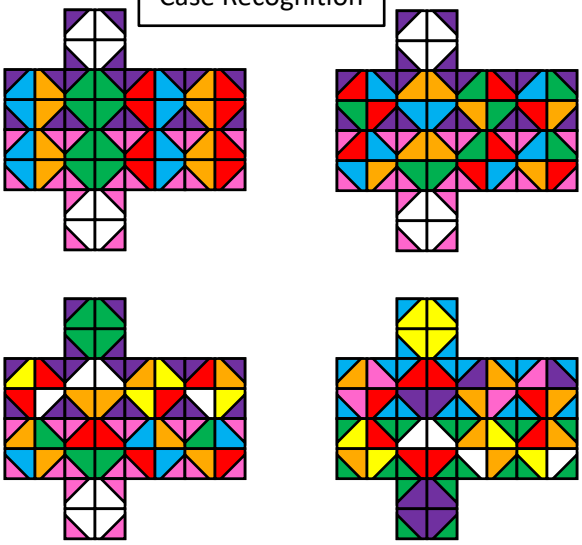
S-S-S-S (1/1296)
Solved





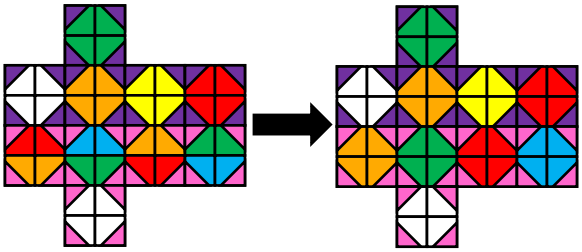
Permutation Parity (1/2)
- U ( D2 R2 U2 F' D2 L2 U2 F' )
- U ( [D2 R]6 )
- R I U2 R I' R U R U2 F I F U2 R U R

## Case Recognition



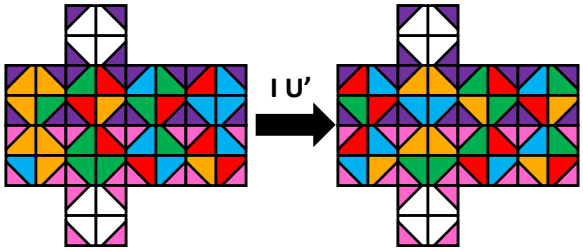
All four states are equivalent, because the bars are on the same side. It does not matter the color of the bar or what the other colors are in relation to it, only that the bars are correct in orientation to each other.

## Permutation Parity

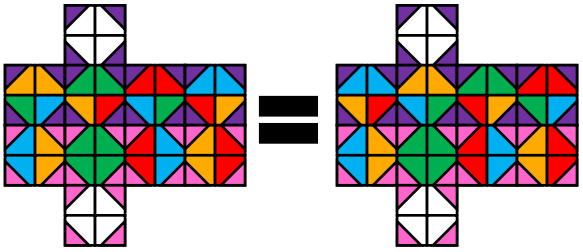


After execution of the PAL algorithm, 50% of the time an I move is sufficient to solve both halves of the cube separately. The other 50% of the time an algorithm is needed to rotate one full layer twice. This algorithm is essentially rotating the buffer layer twice by scrambling and solving the cube in 18 quarter turns. This is fastest done with **D2 R2 U2 F D2 L2 U2 F** on the half of the cube that is completely solved. After this the cube can be solved in a complete reorientation of the U face.

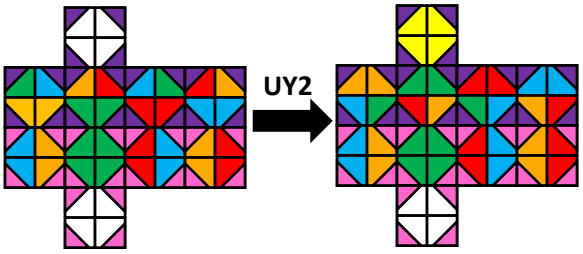
## Case Manipulation




These two states are equivalent because they can the second desired state can be easily reached from the first by an I U' move.



In general, the position of the solved (S) faces and faces without a bar (X) faces do not matter, and further, as long as there is at least one layer without a bar all layers with a bar can be put on the same side. If all layers have a bar then at least three can be put together using only U face and I face moves, and there are different cases for the position of the last bar.




Sometimes it can happen that an X, S or B layer could be switch with another in such a way that it is not possible to get to a described case with just U and I face moves. Then UY2 moves are required to get to specified case, or reorientation of the entire puzzle with an x or z2 reorientation.