

PLATFORM: ALTERNATIVE METHOD LAST LAYER

STEP ONE
STEP TWO

AUTHOR: MUNARO RICCARDO

CODE WCA: 2014MUNA01

INTRODUCTION:

Welcome!

This guide is a alternative solution for LL(Last Layer)

The step one is the creation block 1x1x3.

There are 50 cases and 25 algorythms.

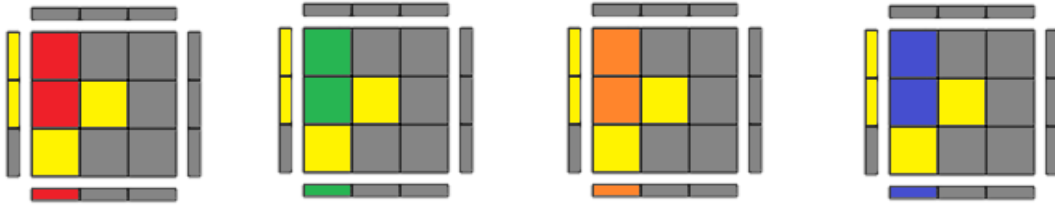
The step two is the solution block 1x1x3.

There are 71 cases and 71 algorythms.

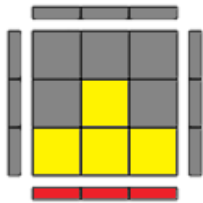
Good vision!

STEP ONE

When you complete first and second layer with cross and F2L you can create a block 1x1x3, a block 1x1x3 can have four colors:



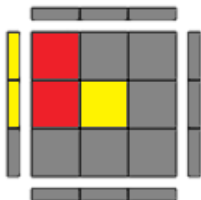
When you use an algorithm you solve block 1x1x3.



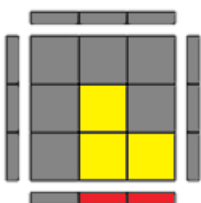
For simpleness I'll use one color.

WARNING:

If in the last layer you don't any case, you can create a parzial block and in the next you can solve a block 1x1x3.
Example:



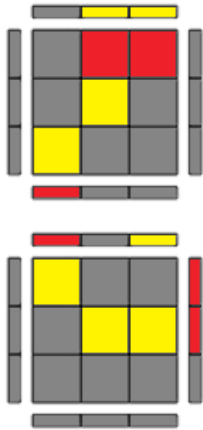
F' L F L' U' L' U L



ANTI-QUANTIK BLOCK LEFT

ALGO: L' U' L U L' F' L' F

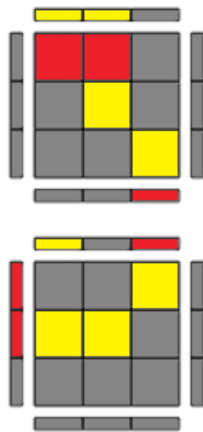
CASES:



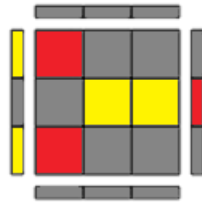
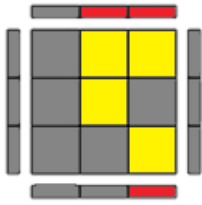
ANTI-QUANTIK BLOCK RIGHT

ALGO: R' U R' U' R' F R F'

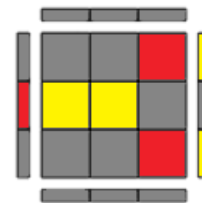
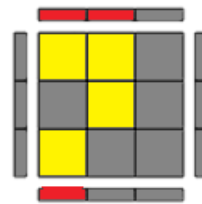
CASES:



ANTI-QUANTIK SIX ANTI-CLOCKWISE
ALGO: F' U' L' U L F
CASES:



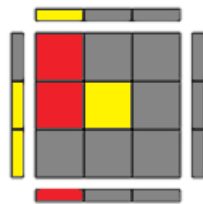
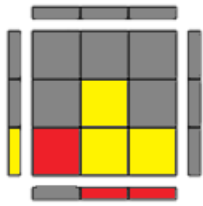
ANTI-QUANTIK SIX CLOCKWISE
ALGO: F U R U' R' F'
CASES:



QUANTIK DOUBLE LEFT

ALGO: L' U' L2 F' L' F2 U' F'

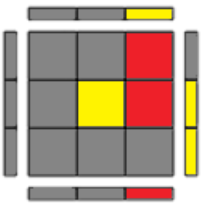
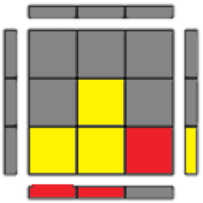
CASES:



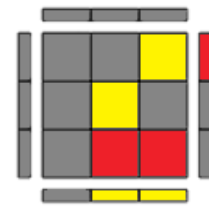
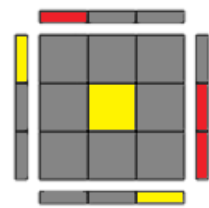
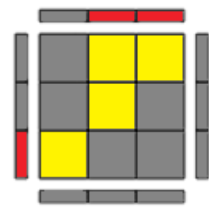
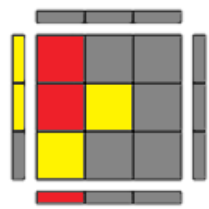
QUANTIK DOUBLE RIGHT

ALGO: R U R2 F R F2 U F

CASES:



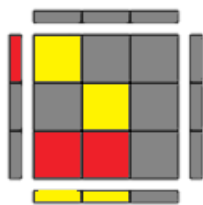
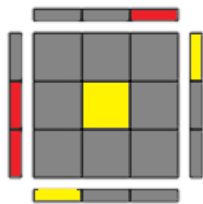
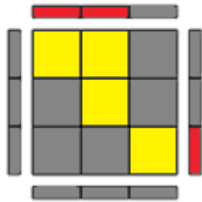
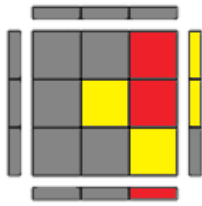
QUANTI-K BLOCK ANTI-CLOCKWISE
ALGO: F' L F L' U' L' U L
CASES:



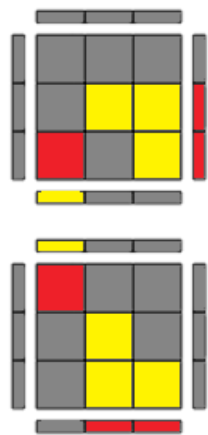
QUANTI-K BLOCK CLOCKWISE

ALGO: F R' F' R U R' U' R'

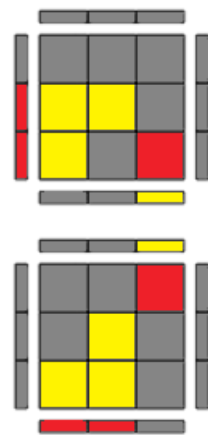
CASES:



QUANTI K EIGHT LEFT
ALGO: L F R' F' L' F R F'
CASES:



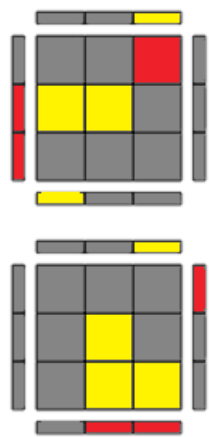
QUANTI K EIGHT RIGHT
ALGO: R' F' L F R F' L' F
CASES:



QUANTIK NORMAL LEFT

ALGO: L' U R U' L U R'

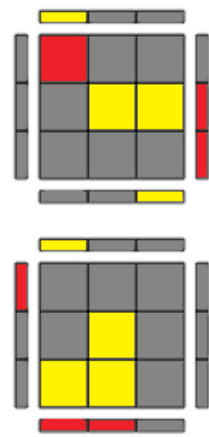
CASES:



QUANTIK NORMAL RIGHT

ALGO: R U' L' U R' U' L

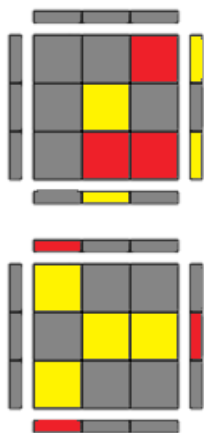
CASES:



QUANTIK SIX ANTI-CLOCKWISE

ALGO: F' L' U' L U F

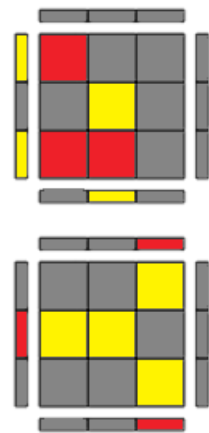
CASES:



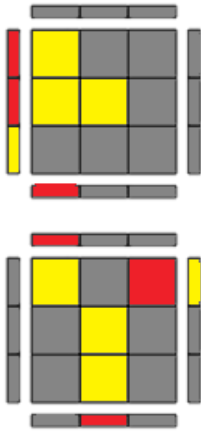
QUANTIK SIX CLOCKWISE

ALGO: F R U R' U' F'

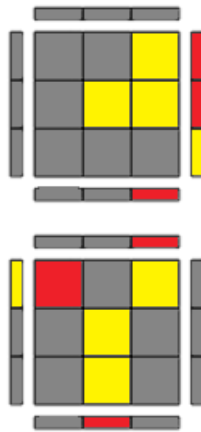
CASES:



QUANTI-K SUN LEFT
 ALGO: L' U2 L U L' U L
 CASES:



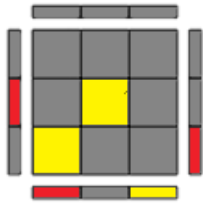
QUANTI-K SUN RIGHT
 ALGO: R U2 R' U' R U' R'
 CASES:



ANTI-QUANTI-K ANGLE LEFT

ALGO: L' U2 L U2 L F' L' F

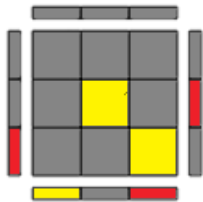
CASES:



ANTI-QUANTI-K ANGLE RIGHT

ALGO: R U2 R' U2 R' F R F'

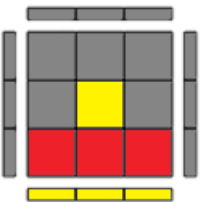
CASES:



INVERSE MEGA CLOCKWISE

ALGO: F R U' B U B' R' F'

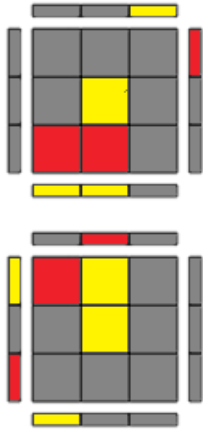
CASES:



QUANTI-K ANGLE ANTI-CLOCKWISE

ALGO: F' L F L' U2 L' U2 L

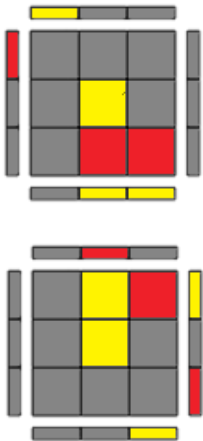
CASES:



QUANTI-K ANGLE CLOCKWISE

ALGO: F R' F' R U2 R U2 R'

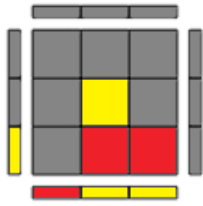
CASES:



QUANTI-K CONSTRUCT ANTI-CLOCKWISE

ALGO: F' L' U B' U B L F U' R U' R'

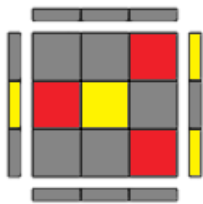
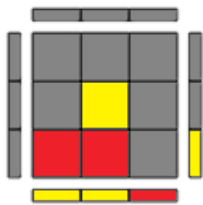
CASES:



QUANTI-K CONSTRUCT CLOCKWISE

ALGO: F R U' B U' B' R' F' U' L' U L

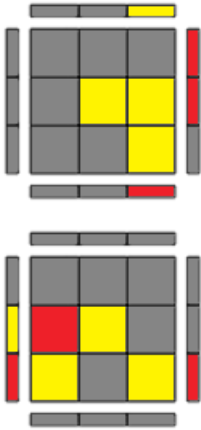
CASES:



SUPER MEGA ANTI-CLOCKWISE

ALGO: F' L' B' U B U' L F

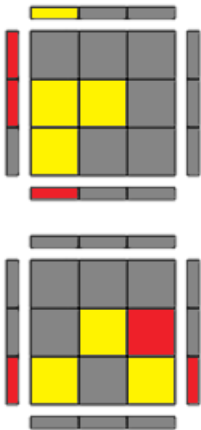
CASES:



SUPER MEGA CLOCKWISE

ALGO: F R B U' B' U' R' F'

CASES:



STEP TWO

This step is very difficult.

I'll use all the colors to represent blocks 1x1x3:

red,green,orange,blue.

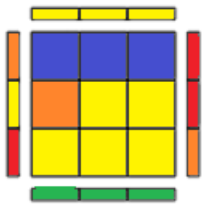
You have to learn how to recognize the cases.

Rare cases are PLL.

P2 LEFT

ALGO: L F U' F' L' B' U2 F U' B U F'

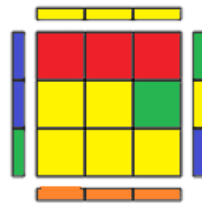
CASE:



P2 RIGHT

ALGO: R' F' U F R B U2 F' U B' U' F

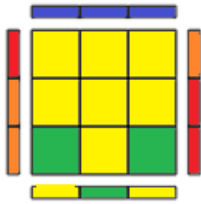
CASE:



PR

ALGO: R2 D R' U2 R D' R' U' R' U' R' F R2 U' R' U' R U
R' F'

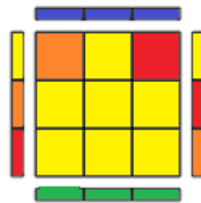
CASE:



ER (OLL24+ U+ T PERM)

ALGO: r U R' U' r' F R F' U R U R' U' R' F R2 U' R' U' R U
R' F'

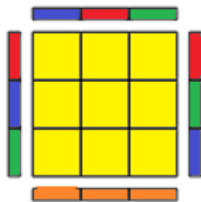
CASE:



F PERM (PLL)

ALGO: R' U R U' R2 F' U' F U R F R' F' R2

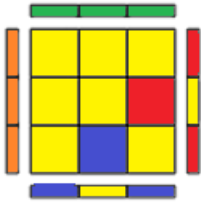
CASE:



2-FLIP ADJACENT (ELL)

ALGO: R B M' U M' U M' U2 M U M U M U2 B' R'

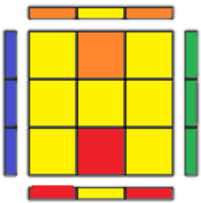
CASE:



2-FLIP OPPOSITE (ELL)

ALGO: M' U M' U M' U2 M U M U M

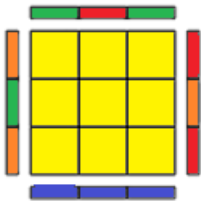
CASE:



Ua PERM (PLL)

ALGO: M2 U M' U2 M U M2

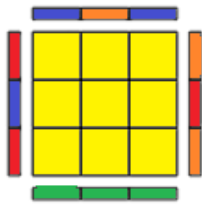
CASE:



U_b PERM (PLL)

ALGO: M² U' M' U² M U' M²

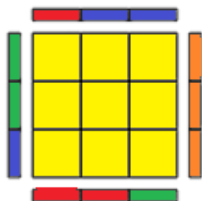
CASE:



J_a PERM (PLL)

ALGO: L' U' L F L' U' L U L F' L² U L

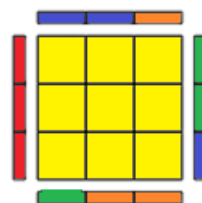
CASE:



J_b PERM (PLL)

ALGO: R U² R' U' R U² L' U R' U' L

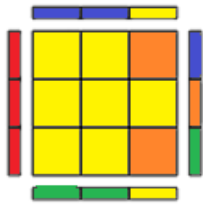
CASE:



A14

ALGO: L' U' L U' L' U2 L R U R' U R U2 R'

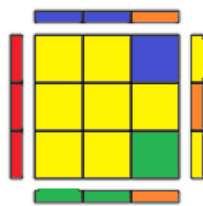
CASE:



E14

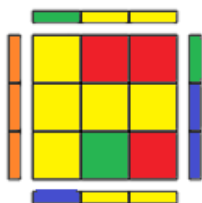
ALGO: R U2 R' U' R U' L' R' U2 L U L' U L

CASE:



QU ANTI-CLOCKWISE

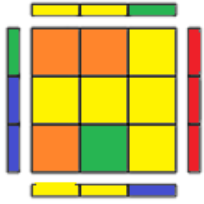
ALGO: F' L F L' U' L' U L2 F' L' F U F U' F'



QU CLOCKWISE

ALGO: F R' F' R U R U' R² F R F' U' F' U F

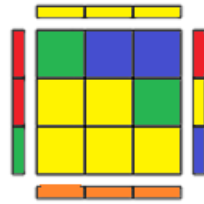
CASE:



A-QU LEFT

ALGO: L' U' L U L F' L' F² U F' U' F' L F L'

CASE:



A-QU RIGHT

ALGO: R U R' U' R' F R F² U' F U F R' F' R

CASE:



K LEFT

ALGO: M L' U² L U L' U² R U' L U R' M'

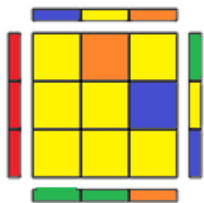
CASE:



K RIGHT

ALGO: M R U² R' U' R U² L' U R' U' L M'

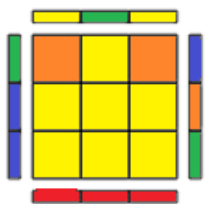
CASE:



S2E LEFT

ALGO: M' U' M U² M' U' M L F U' F' L' B' U² F U' B U F'

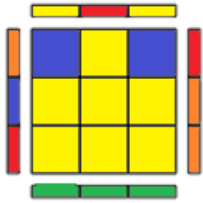
CASE:



S2E RIGHT

ALGO: M' U M U2 M' U M R' F' U F R B U2 F' U B' U' F

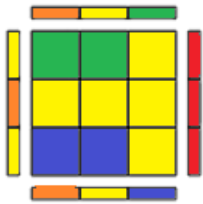
CASE:



DD2

ALGO: R U R2 F R' F2 U F L' U' L2 F' L' F2 U' F'

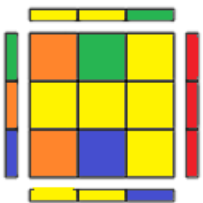
CASE:



TR

ALGO: R U R' U' R' F R F' U' R' U2 R U2 R' F R U R' U'
R' F' R2

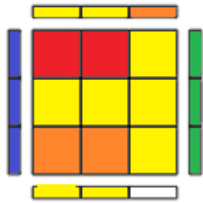
CASE:



T0

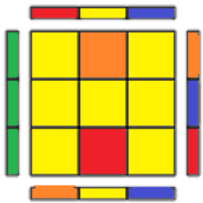
ALGO: R U R' U' R' F R F' U R U² R' U' R U² L' U R' U'
L

CASE:



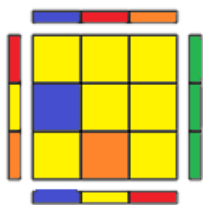
T2

ALGO: R U² R' U' R U² L' U R' U' B' R B R L U' R' U R
CASE:



I-23 RIGHT

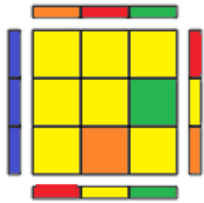
ALGO: R' F' L F R L' U' L' U' L U L' U² R U' L U R'
CASE:



I-23 LEFT

ALGO: L F R' F' L' R U R U R' U' R U² L' U R' U' L

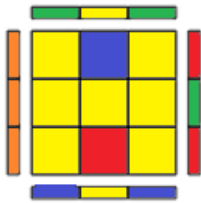
CASE:



S2 RIGHT

ALGO: R U R' U' M' U R U' r'

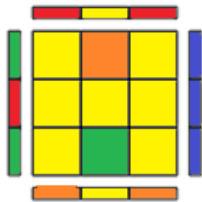
CASE:



S2 LEFT

ALGO: L' U' L U M' U' L' U¹

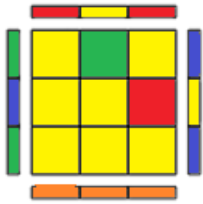
CASE:



S3 RIGHT

ALGO: R' F' L' U2 L F R B U2 B'

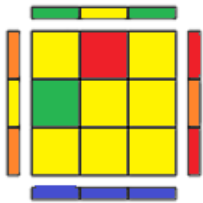
CASE:



S3 LEFT

ALGO:

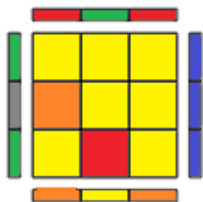
CASE:



SQ RIGHT

ALGO: R' F' L F R L' U' L' U L

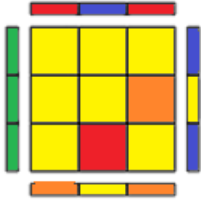
CASE:



SQ LEFT

ALGO: L F R' F' R L' U R U' R'

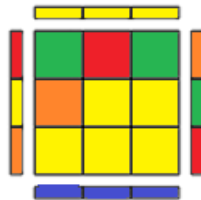
CASE:



Q14 LEFT

ALGO: L F L' B L F' L2 B' L U L U' L'

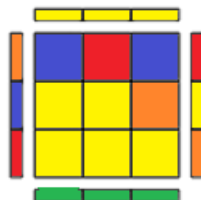
CASE:



Q14 RIGHT

ALGO: R' F' R B' R' F R2 B R' U' R' U R

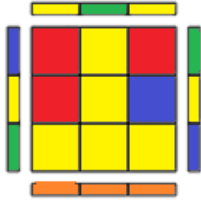
CASE:



A-Q14 ANTI-CLOCKWISE

ALGO: F' U' F U F R' F2 L' F R F' L F

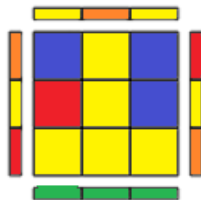
CASE:



A-Q14 CLOCKWISE

ALGO: F U F' U' F' L F2 R F' L' F R' F'

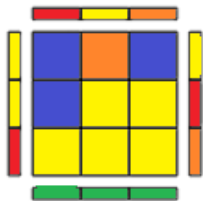
CASE:



QMA ANTI-CLOCKWISE

ALGO: B L' B' R B L B' L' R' B L B' U2 B' U2 B

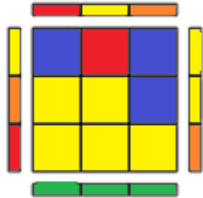
CASE:



QMA CLOCKWISE

ALGO: B' R B L' B' R' B R L B' R' B U2 B U2 B'

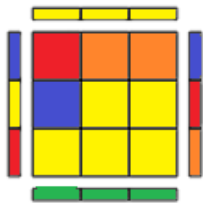
CASE:



A-QMA CLOCKWISE

ALGO: F U2 F' U2 F' L F R' L' F' L F R F' L' F

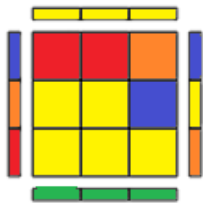
CASE:



A-QMA ANTI-CLOCKWISE

ALGO: F' U2 F U2 F R' F' L R F R' F' L' F R F'

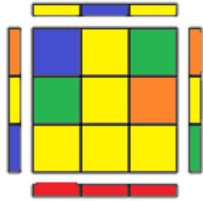
CASE:



QXT ANTI-CLOCKWISE

ALGO: F' L F L' U' L' U² R U' L U R² F R F' U' F' U F

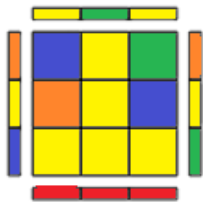
CASE:



QXT CLOCKWISE

ALGO: F R' F' R U R U² L' U R' U' L² F' L' F U F U' F'

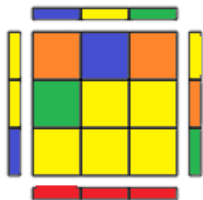
CASE:



A-QXT LEFT

ALGO: L U L' U' L' B L B² U F U' B U² F' U' F' L F L'

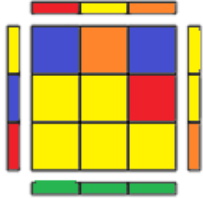
CASE:



A-QXT RIGHT

ALGO: R' U' R U R B' R' B2 U' F' U B' U2 F U F R' F' R

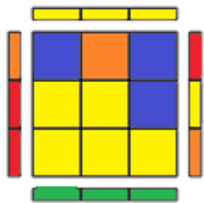
CASE:



P23 CLOCKWISE

ALGO: B L' B' L U L U' L2 B L B' U' B' U B R B' R' B U
B U' B'

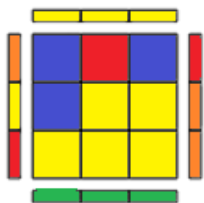
CASE:



P23 ANTI-CLOCKWISE

ALGO: B' R B R' U' R' U R2 B' R' B U B U' B' L' B L B' U'
B' U B

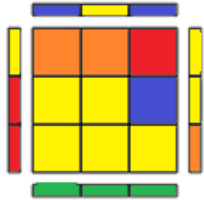
CASE:



LU LEFT

ALGO: L F R U' R' U² F' U' R L' U R' U' R' F R F'

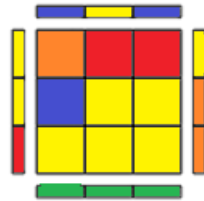
CASE:



LU RIGHT

ALGO: R' F' L' U L U² F U L' R U' L U L F' L' F'

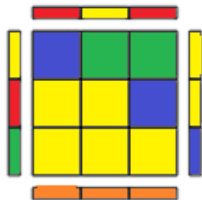
CASE:



H22 LEFT

ALGO: L' U' R U R' U' L U R U' R' L' U L B' R B R' U' R'
U R

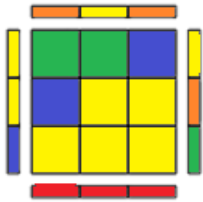
CASE:



H22 RIGHT

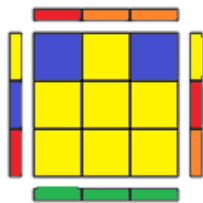
ALGO: R U L' U' L U R' U' L' U L R U' R' B L' B' L U L U'
L'

CASE:



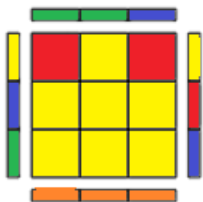
QE LEFT

ALGO: L F' L' U' F' U F U' R' L F2 L' R F' L F L' U' L' U L
CASE:



QE RIGHT

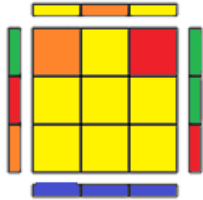
ALGO: R' F R U F U' F' U L R' F2 R L' F R' F' R U R U' R'
CASE:



MIX RIGHT

ALGO: U' R' F R F' U' F' U F R B U B' U' R² U L U' R U
L'

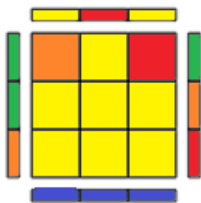
CASE:



MIX LEFT

ALGO: U L F' L' F U F U' F' L' B' U' B U L² U' L' U L' U'
R

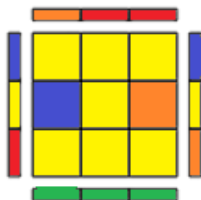
CASE:



H LEFT:

ALGO: L U L' U B' R' F' U² F R U F U' B U F'

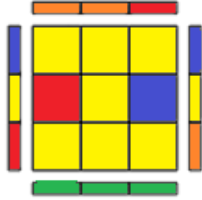
CASE:



H RIGHT

ALGO: R' U' R U' B L F U² F' L' U' F' U B' U' F

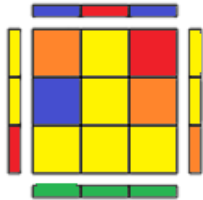
CASE:



AH5 LEFT

ALGO: L U L' U' L' U' L U L U' L' U' L' U L B U B' U' B' R B R'

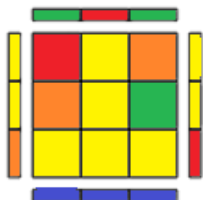
CASE:



AH5 RIGHT

ALGO: :R' U' R U R U R' U' R' U R U R U' R' B' U' B U B L' B' L

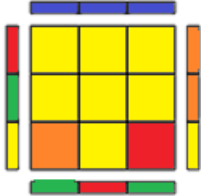
CASE:



LE RIGHT

ALGO: R U2 R' U' R' F R F2 U F L F' L' F U F U' F'

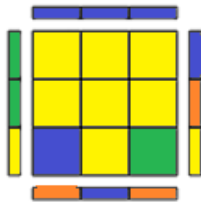
CASE:



LE LEFT

ALGO: L' U2 L U L F' L' F2 U' F' R' F R F' U' F' U F

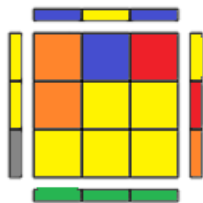
CASE:



A-11 CLOCKWISE

ALGO: B U B2 R B R' U R B' R' B L' B' R' U R U' B L

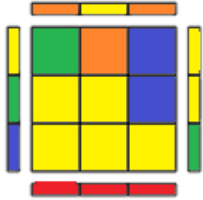
CASE:



A-11 ANTI-CLOCKWISE

ALGO: B' U' B2 L' B' L U' L' B L B' R B L U' L' U B' R'

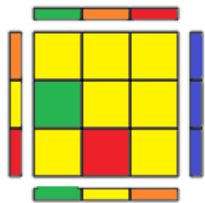
CASE:



S2A LEFT

ALGO: l' U' L U M U' L' U' L U L' U2 R U' L U' R'

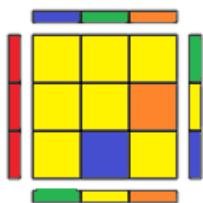
CASE:



S2A RIGHT

ALGO: r U R' U' M U R U R' U' R U2 L' U R' U' L

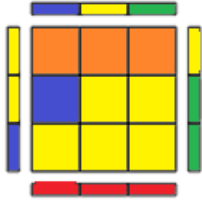
CASE:



QUU RIGHT

ALGO: R U R' U R U2 L' R' U2 L U' L' U' L2 F' L F2 U' F'

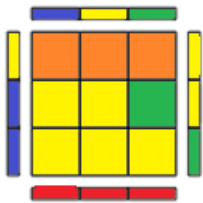
CASE:



QUU LEFT

ALGO: L' U' L U' L' U2 L R U R' U R U' R2 F R' F2 U F

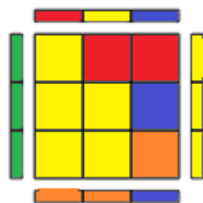
CASE:



PR4 RIGHT

ALGO: R U R2 F R' F2 U F U L' U2 L U L' U R L U2 R'
U' R U' R

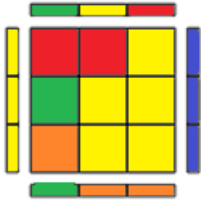
CASE:



PR4 LEFT

ALGO: L' U' L2 F' L F2 U' F' U' R U2 R' U' R U' L' R' U2
L U L' U L

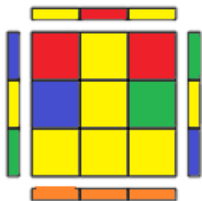
CASE:



T4

ALGO: F' L' B' U B L F R M' U M U2 M' U M R' F' U F R
B U2 F' U B' U' F

CASE:



CONCLUSION

I created this method since 2008, It was difficult create the guide.