

## PPA method

## Overview

Step 1: Corners of white layer (CWL)

Step 2: Build the 2x2x3 at Front (or anywhere)

Step 3: Build the 1x2x3 at Back (or any where but must opposite the 2x2x3 block)

Step 4: Corners of last layer (CMLL)

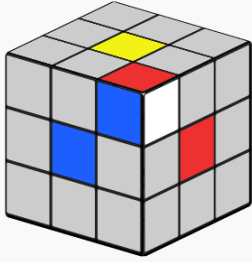
Step 5: Edge of yellow layer (EYL)

Step 6: Permutation Edge of last layer (PELL)

Step 1: CWL

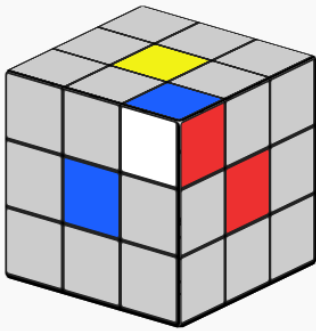
We have 3 cases:

Case 1:



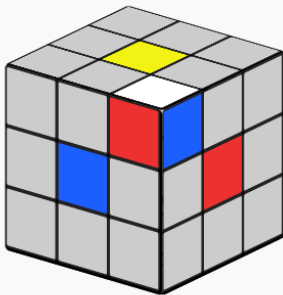
algorithms:  $R\ U\ R'$

Case 2:



algorithms:  $U\ R\ U'\ R'$

Case 3:



algorithms:  $R\ U^2\ R'\ U'\ R\ U\ R'$

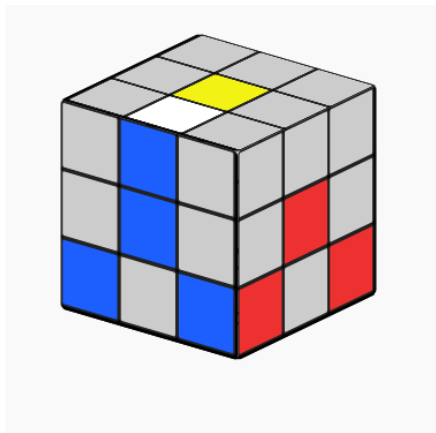
\*Special case: If the corner at the white layer, use  $R\ U\ R'$  or  $L'\ U'\ L$  and consider the case (case 1; 2 or 3)

Step 2: Build a 2x2x3 at Front (Red) (or anywhere)

Well, this step is self-experimentation, but I can help you.

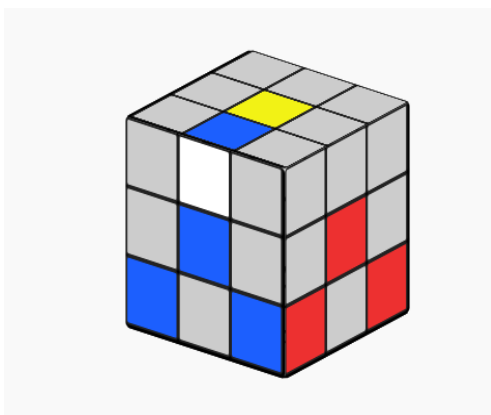
First, solve the white edge, we have 2 cases:

Case 1:



algorithms:  $M'\ U^2\ M$

Case 2:

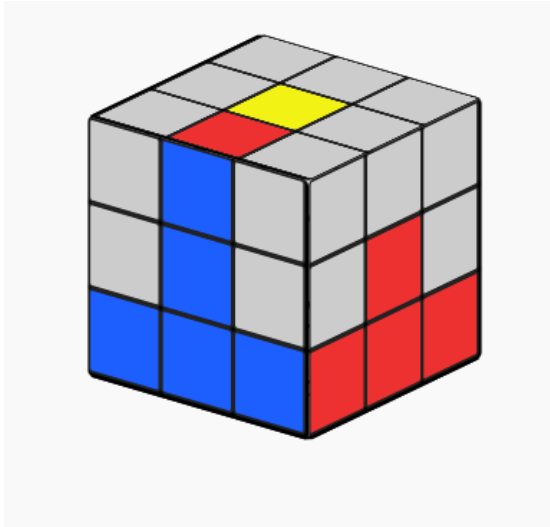


algorithms:  $U\ M'\ U'\ M$

\*Special case: If Edge at the white layer, algorithms:  $M'\ U\ M$  and consider the case.

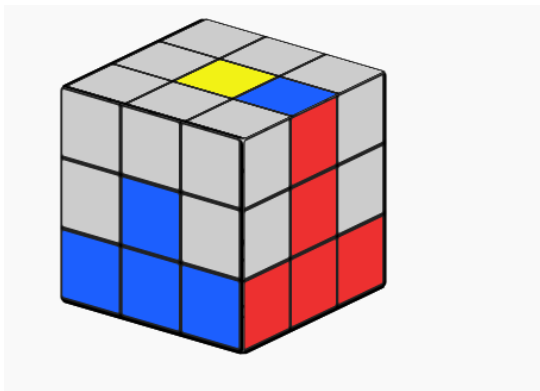
Next, build the 2x2x3 block. Don't worry, just 3 cases at LBL method step 2.

Cases 1:



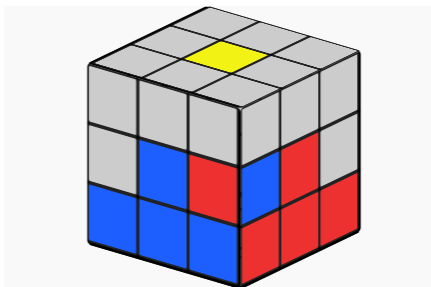
Algorithms:  $U R U R' U' F' U$

Cases 2:



Algorithms:  $U' L' U L U F U' F'$

Cases 3:



Algorithms:  $R^2 U^2 F R^2 F' U^2 R' U R'$

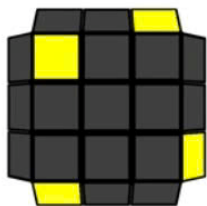
Step 3: Build the 1x2x3 at Back (or opposite the 2x2x3 block)

Well, this step like the step 2 but you do the opposite the 2x2x3 block (also same algorithms)

Step 4: CMLL

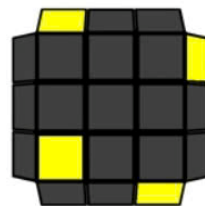
Well, you don't need to learn all 41 algorithms of CMLL, you only need learn 2look CMLL

This is 7 algorithms of 2look CMLL (The turns in “()” mean you can do the finger trick with)



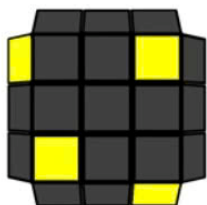
**Anti-Sune**

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



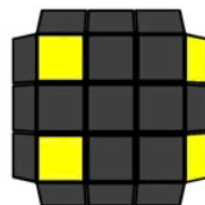
**Sune**

R U R' U R U2 R'



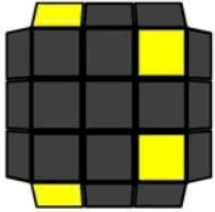
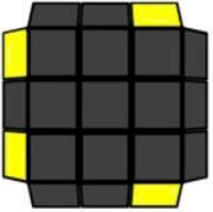
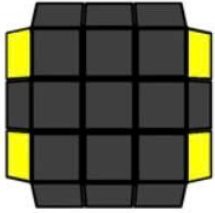
**L/ Bowtie**

[U'] F R U' R' U' (R U R' F')



**U/ Headlights**

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

|                                                                                                                                                                      |                                                                                                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p><b>T/ Chameleon</b></p> <p><math>(R\ U\ R'\ U')\ R'\ F\ R\ F'</math></p>        |  <p><b>Pi/ Dead Guy</b></p> <p><math>F\ (R\ U\ R'\ U')^2\ F'</math></p> |
|  <p><b>H/ Double Anti-Sune</b></p> <p><math>[U]\ F\ (R\ U\ R'\ U')^3\ F'</math></p> |                                                                                                                                                            |

(Image source: <https://rubikonline.vn/roux-method>)

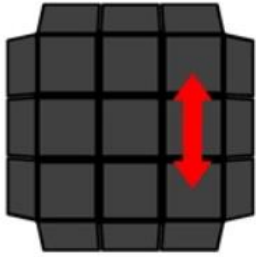
After do this, you need to permute the corner of yellow layer. And we have 2 cases:

Case 1:



Algorithms:  $F\ R\ U'\ R'\ U'\ (R\ U\ R'\ F')$   
 $(R\ U\ R'\ U')\ R'\ F\ R\ F'$

Case 2:

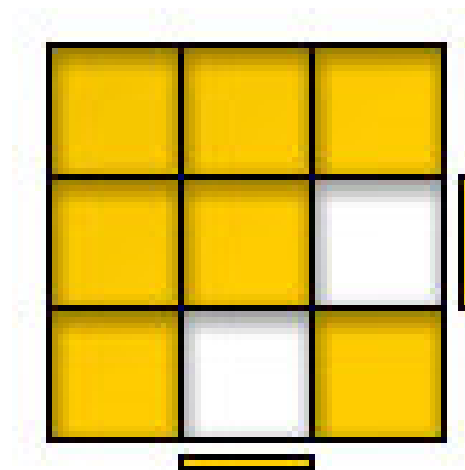


Algorithms:  $(R \ U \ R' \ F')$   $(R \ U \ R' \ U')$   
 $R' \ F \ R^2 \ U' \ R'$

Step 5: EYL

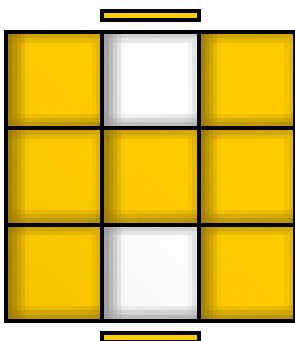
Only 3 cases:

Case 1:



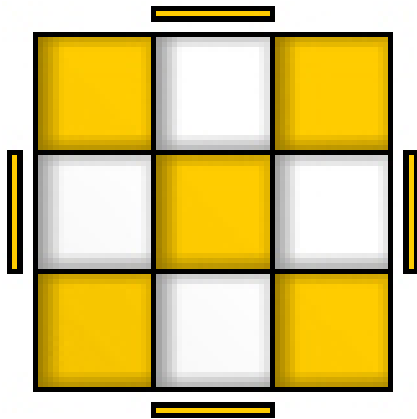
Algorithms:  $M \ U \ M' \ U^2 \ M \ U \ M'$

Case 2:



Algorithms:  $x \ M' \ U' \ R \ U \ M \ U' \ R' \ U' \ x'$

Case 3:



Algorithms:  $M' U^2 M U^2 M' U M U^2$   
 $M' U^2 M$

Does this look familiar? Yes, this is OLL but I choose the easiest to remember algorithms.

Step 6: PELL:

This is the last step, it's only PLL  $U_a$ ,  $U_b$ , H and Z. (The turn in “()” mean you can do with finger trick)

|       |  |                                                                                                                                                                                                                   |
|-------|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $U_a$ |  | <ul style="list-style-type: none"> <li>▪ <math>(R U' R U) R U (R U' R' U') R^2</math></li> <li>▪ <math>M^2 U (M U^2 M') U M^2</math></li> <li>▪ <math>y^2 (R^2 U' R' U') R U R U (R U' R)</math></li> </ul>       |
| $U_b$ |  | <ul style="list-style-type: none"> <li>▪ <math>R^2 U (R U R' U') R' U' (R' U R')</math></li> <li>▪ <math>M^2 U' (M U^2 M') U' M^2</math></li> <li>▪ <math>y^2 (R' U R' U') R' U' (R' U R U) R^2</math></li> </ul> |
| Z     |  | <ul style="list-style-type: none"> <li>▪ <math>(M^2' U M^2' U) M' U^2 (M^2' U^2 M')</math></li> <li>▪ <math>y' M' U (M^2' U M^2') U (M' U^2 M^2)</math></li> </ul>                                                |
| H     |  | <ul style="list-style-type: none"> <li>▪ <math>(M^2' U M^2') U^2 (M^2' U M^2')</math></li> </ul>                                                                                                                  |

Image source: <https://rubikonline.vn/21-cong-thuc-pll-hoan-vi-tang-cuoi-cung-cfop>

## SUMMARY & CONCLUSION

### PPA METHOD IN NUMBERS:

- Total algorithms:24
- Steps: 6
- Intuitive steps:3 (Steps 2-3)
- Algorithm steps:3 (Steps 4-6)

### WHO IS PPA FOR?

- CFOP beginners:struggling with F2L algorithms
- Intuitive solvers: who prefer building over memorization
- Method explorers:looking for new approaches
- Casual cubers:wanting a complete method with minimal algs

### LEARNING ADVANTAGES:

1. Gradual learning: Start intuitive, add algs later
2. Less overwhelm: Only 24 algs vs 120+ (CFOP)
3. Clear structure: 6-step process, easy to follow
4. Transferable skills: Block building helps other methods

### PERFORMANCE POTENTIAL:

- Current PB: 44.094 seconds (with \$2 cube)
- Estimated potential: Sub-30 achievable
- Movecount:~55-60 moves (optimization possible)
- Ergonomics: Good - uses R, U, M moves mostly
- NEXT DEVELOPMENT STEPS:

1. Create video tutorial series
2. Develop advanced PPA variants
3. Optimize movecount and recognition
4. Build PPA community and resources

#### - FINAL THOUGHTS:

PPA Method offers a viable alternative for cubers who find traditional methods too algorithm-heavy. By combining corner-first intuition with efficient block building and a simplified last layer, PPA provides a complete solving system with significantly fewer algorithms to memorize.

While not designed to surpass CFOP in raw speed, PPA excels in accessibility and learning efficiency - making it an excellent choice for beginners and those who value intuitive solving over pure memorization.

PPA Method - Made by Phan Phước An, January 2026