

A Method to Guarantee Solving all 8! Permutations of Wing Edges in 3 or Less Algorithms.

By: Christopher Mowla (cmowla)

Assuming that one is using K4 to solve all of a big cube except for the last layer wing edges,

Step 1: Solve one dedge.

Step 2: OLL of Three or Less Remaining Unsolved Dedges

Step 3: PLL of Three or Less Remaining Unsolved Dedges

- Number of algorithms to be learned (including mirrors, inverses, and related algorithms): 66.
- Number of algorithms excluding mirrors and inverses: 31.
- Number of algorithms excluding mirrors, inverses and little differences in the algorithms (how many distinct algorithms were required to make all 66 cases): 18-20 (depending on whether or not you choose to use the general algorithm for all 3-cycles and 2 3-cycles shown on the last page).

Cycles

2-Cycles: 4 (For PLL)

3-Cycles: 16 (12 for solving the first dedge and OLL of one dedge, 4 for PLL)

2 2-Cycles: 29 (For solving the first dedge, OLL, and PLL)

1 3-cycle and 1 2-cycle: 12 (For PLL)

2 3-cycles: 4 (For PLL)

3 2-cycle: 1 (For OLL of 3 Dedges)

Total = 66

Average moves (in btm) of lowest move count algorithms provided for the 66 cases (rounded to the 2nd decimal place).

2-Cycles: 15.00

3-Cycles: 10.00

2 2-Cycles: 11.48

1 3-cycle and 1 2-cycle: 17.33

2 3-cycles: 11.50

3 2-cycle: 19.00

Total Average: 12.52

The total average is not the average of the averages above. Rather, the average of the lowest move count algorithms for all 66 cases.

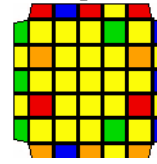
Notes

- The moves M and E mean all inner layer slices, not just the central slice on odd cubes.
- For those who don't know, an algorithm of the form [A: B] = A B A'.
- Above each of the three steps, I put an ordered triple (A, B, C). 'A' means the total number of cases, 'B' means the total number of cases minus the mirrors and inverses, and 'C' means the number of algorithms I used/modified slightly to make all of the algorithms.
- The longest 2 2-cycle is 13 btm (Step 1), the longest OLL is 19 btm (Step 2), and the longest PLL is 18 btm (Step 3); therefore establishing an upper bound of 50 moves for any case. However, it's safe to say that even the worst cases can be handled with around 40 moves because the averages for steps 1-3 three are $10.81 + 11.37 + 13.44 = 35.62$.

Step 1: Solving the First Dedge

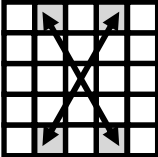
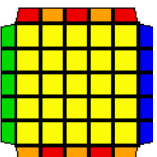
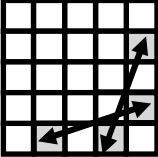
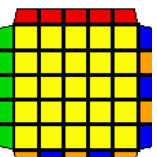
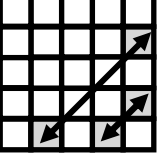
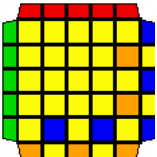
Step 1A

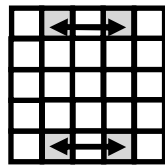
Prerequisite: None of the last layer dedges are solved (even half way solved). Example



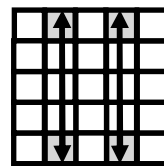
- Use one of the 41 algorithms in this section to solve the dedge in one algorithm.
- Although there are technically 41 different algorithms needed, all of them can be formed from only 11 distinct algorithms. The cases have been grouped into sections to indicate algorithm relationships.

2 2-Cycles (29, 19, 8)

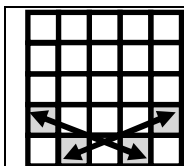
 [x' Rw2 U2 F2: r2] (7) [Rw2 F2 U2: r2] (7)	
 [F2 U : [Rw2 U2 F2: r2]] (11) [R2 D' x: [Rw2 F2 U2: r2]] (11) [Rw' U R U Lw' U2 Rw' U2: r2] (17)	
 [(R x') U: [Rw2 U2 F2: r2]] (11)	



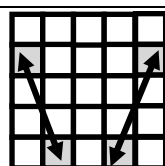
l' U2 l2 F2 r' D2 r' D2 r2 F2 l' U2 (12)
U2 r U2 r2 x U2 r U2 l U2 l' r' U2 r x' (13)
r U2 l' r' U2 r U2 r' U2 l r U2 r' U2 (14)



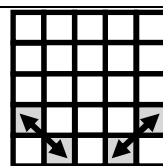
F2 l E F2 E' l' r' E F2 E' r F2 (12)
r U2 l D2 l' U2 (l r') U2 r D2 r' U2 l' (14)



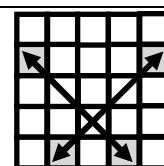
[r2 B2 U l2 U': l2] (11)



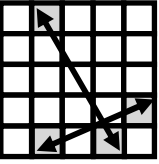
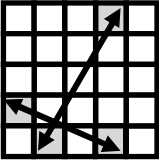
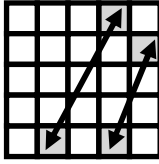
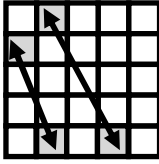
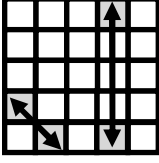
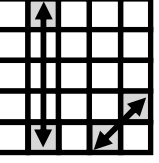
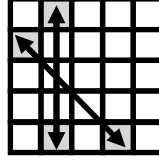
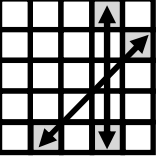
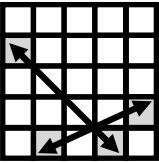
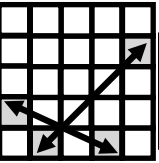
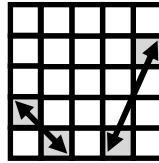
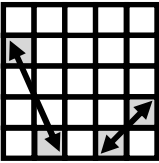
[r2 B2 U' l2 U: l2] (11)

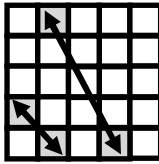
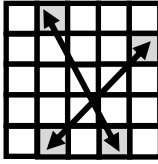
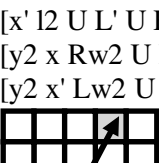
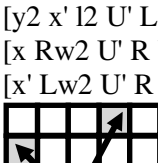
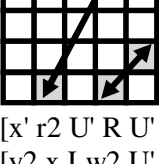
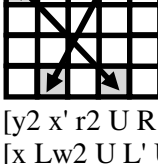
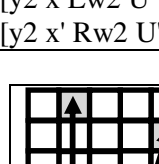
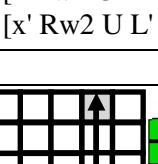


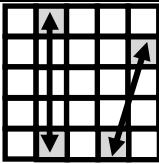
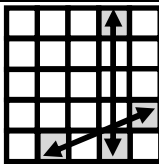
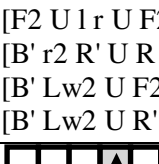
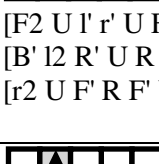
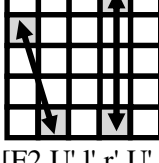
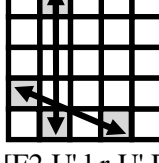
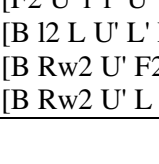
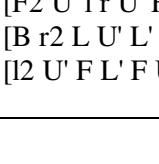
[r2 U2 F M' U: r2] (11)



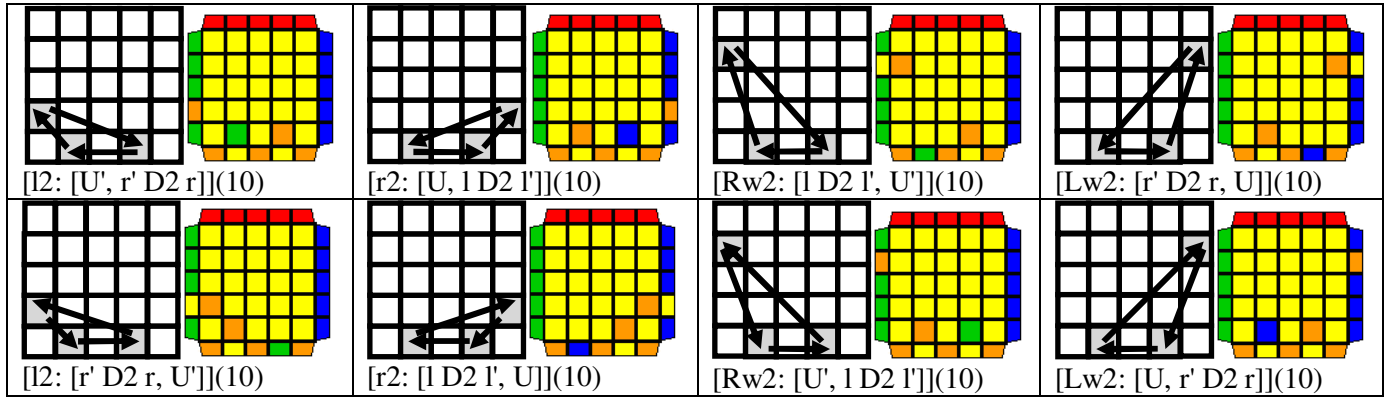
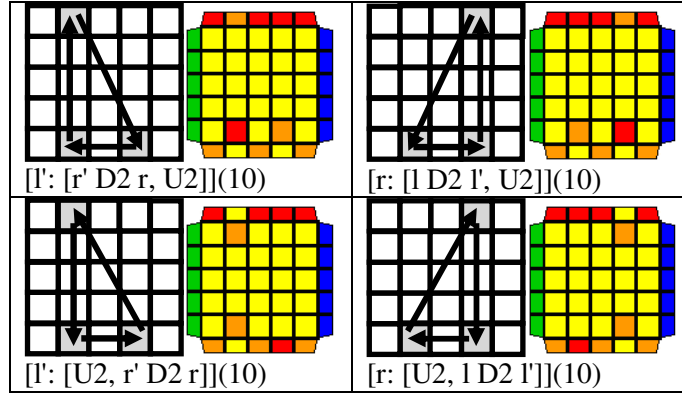
[r2 U2 F' M' U': r2] (11)

 [r2 U F2 U: r2] (9)	 [l2 U' F2 U': l2] (9)	 [l2 U F2 U: l2] (9)	 [r2 U' F2 U': r2] (9)
 [y L F: [r2 U F2 U: r2]] (13)	 [y' R' F': [l2 U' F2 U': l2]] (13)	 [y L F: [l2 U F2 U: l2]] (13)	 [y' R' F': [r2 U' F2 U': r2]] (13)
 [L' B': [r2 U F2 U: r2]] (13)	 [R B: [l2 U' F2 U': l2]] (13)	 [L' B': [l2 U F2 U: l2]] (13)	 [R B: [r2 U' F2 U': r2]] (13)

 [x' l2 U L' U F2: l2] (11)	 [y2 x' l2 U' L' U' F2: l2] (11)
 [y2 x Rw2 U R U F2: l2] (11)	 [x' Lw2 U' R U' F2: l2] (11)
 [x' r2 U' R U' F2: r2] (11)	 [y2 x' r2 U R U F2: r2] (11)
 [y2 x Lw2 U' L' U' F2: r2] (11)	 [x' Rw2 U L' U F2: r2] (11)

 [F2 U l r U F2 U: l2] (15)	 [F2 U l' r' U F2 U: r2] (15)
 [B' r2 R' U R F2 U: r2] (15)	 [B' l2 R' U R F2 U: l2] (15)
 [B' Lw2 U F2 R' U: l2] (13)	 [r2 U F' R' U: r2] (13)
 [F2 U' l' r' U' F2 U': r2] (15)	 [F2 U' l r U' F2 U': l2] (15)
 [B l2 L U' L' F2 U': l2] (15)	 [B r2 L U' L' F2 U': r2] (15)
 [B Rw2 U' F2 L U': r2] (13)	 [l2 U' F' L' U': l2] (13)
 [B Rw2 U' L F2 U': r2] (13)	

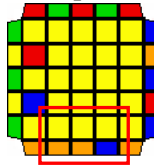
3-Cycles (12, 3, 2)



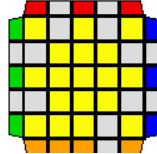
Step 1B

Prerequisite: At least one of the dedges is half way solved.

Example



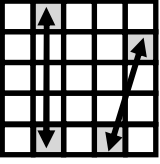
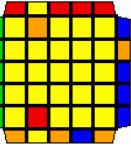
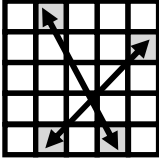
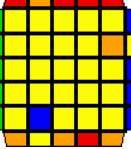
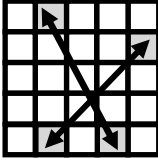
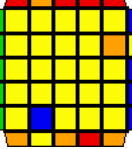
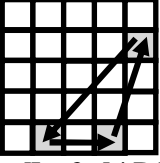
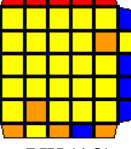
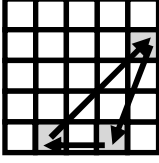
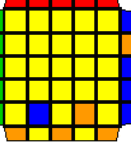
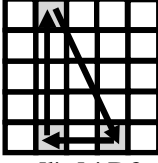
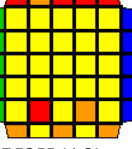
Case 1: The Left Wing Is Solved

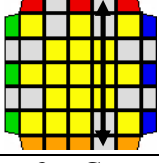
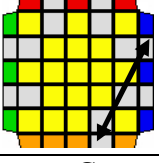
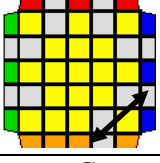
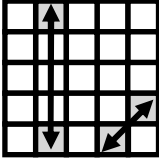
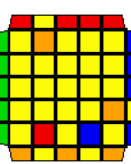
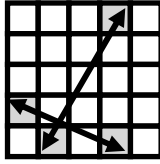
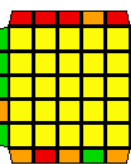
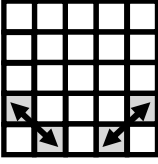
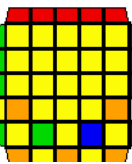
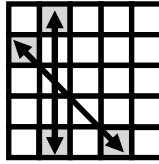
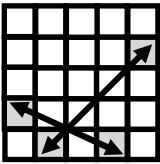
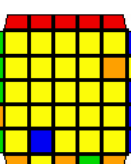
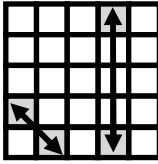
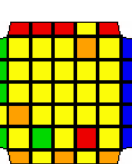
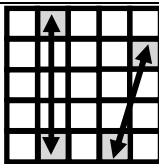
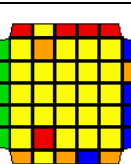
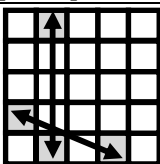
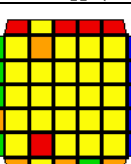
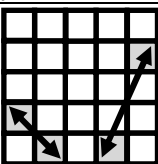
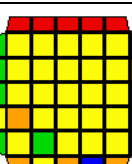
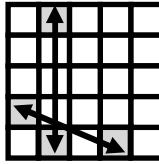
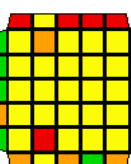
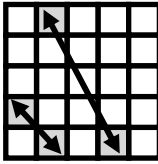
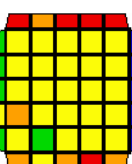
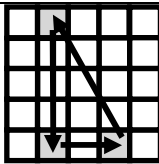
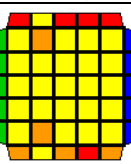
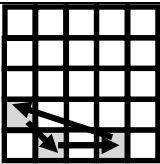
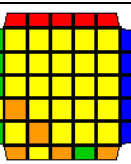
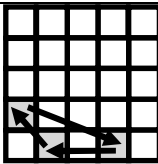
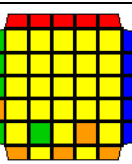


- There are 6 possible cases to solve the right wing.
- We use the 2 2-cycle algorithms shown in Step 1A (excluding the first 5 cases) to solve the remaining right wing edge, with the only restriction that we do not mess up the left wing.
- You can use more than one algorithm for each of the 6 cases.
- I will list all possible ways to handle each of the 6 cases with the 2 2-cycle cases given in the order that the cases appeared in step 1 (not regarding algorithm length).

Also note that you will need to rotate the cube before applying the algorithms.

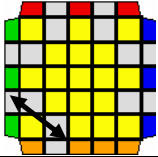
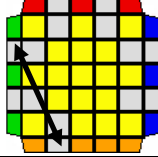
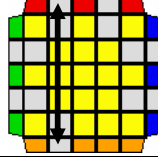
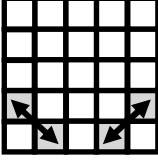
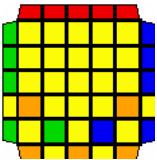
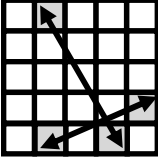
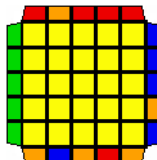
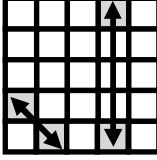
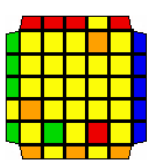
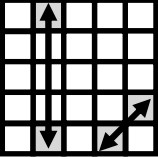
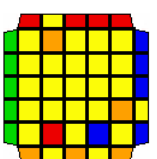
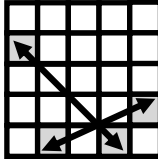
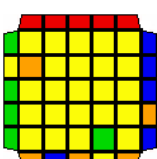
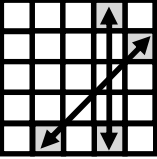
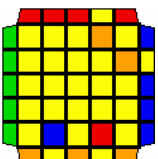
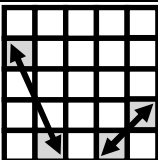
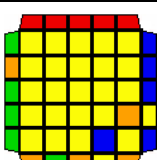
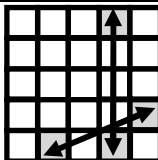
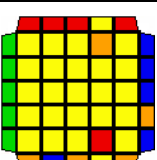
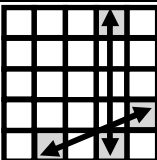
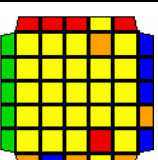
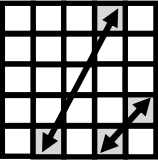
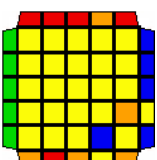
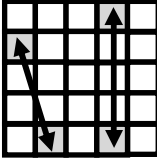
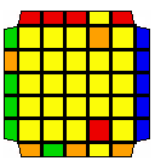
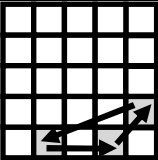
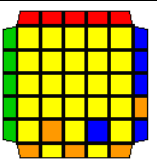
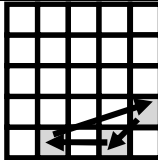
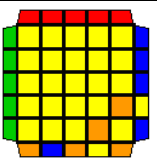
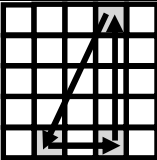
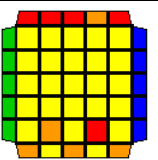
y' + Case	y' + Case	y2 + Case
[r2 B2 U' l2 U: l2] (11)	[r2 U2 F' M' U': r2] (11)	[r2 U F2 U: r2] (9)
[l2 U F2 U: l2] (9)	[y' R' F': [r2 U' F2 U': r2]] (13)	[r2 U' F2 U': r2] (9)
[L' B': [l2 U F2 U: l2]] (13)	[R B: [l2 U' F2 U': l2]] (13)	 [y2 x Rw2 U R U F2: l2] (11) [y2 x' Lw2 U R U F2: l2] (11)

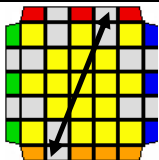
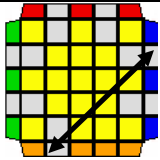
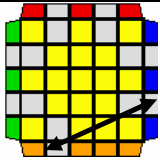
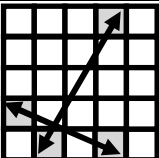
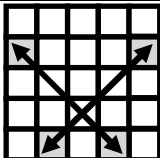
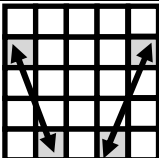
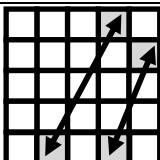
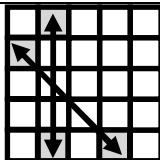
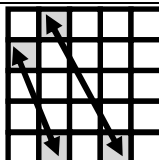
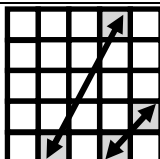
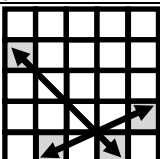
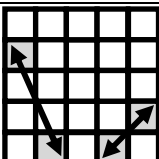
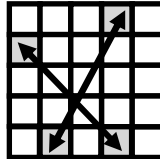
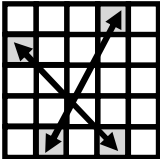
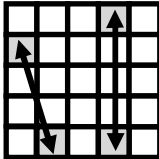
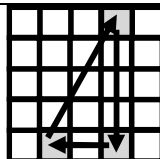
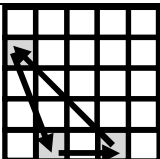
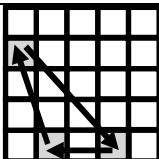
  [F2 U1 r U F2 U: l2] (15) [B' r2 R' U R F2 U: r2] (15) [B' Lw2 U F2 R' U: l2] (13) [B' Lw2 U R' F2 U: l2] (13)	  [y2 x' l2 U' L' U' F2: l2] (11) [x Rw2 U' R U' F2: l2] (11) [x' Lw2 U' R U' F2: l2] (11)	  [y2 x' l2 U' L' U' F2: l2] (11) [x Rw2 U' R U' F2: l2] (11) [x' Lw2 U' R U' F2: l2] (11)
  [Lw2: [r' D2 r, U]](10)	  [Lw2: [U, r' D2 r]](10)	  [l': [r' D2 r, U2]](10)

 y2 + Case	 y + Case	 y + Case
  [y' R' F': [l2 U' F2 U': l2]] (13)	  [l2 U' F2 U': l2] (9)	  [r2 U2 F M' U: r2] (11)
  [y L F: [l2 U' F2 U': l2]] (13)	  [R B: [l2 U' F2 U': l2]] (13)	  [y L F: [r2 U' F2 U': r2]] (13)
  [F2 U1 r U F2 U: l2] (15) [B' r2 R' U R F2 U: r2] (15) [B' Lw2 U F2 R' U: l2] (13) [B' Lw2 U R' F2 U: l2] (13)	  [F2 U'1 r U' F2 U': l2] (15) [B r2 L U' L' F2 U': r2] (15) [l2 U' F L' F U': l2] (13)	  [L' B': [l2 U' F2 U': l2]] (13)
  [F2 U'1 r U' F2 U': l2] (15) [B r2 L U' L' F2 U': r2] (15) [l2 U' F L' F U': l2] (13)		  [x' l2 U L' U F2: l2] (11) [y2 x Rw2 U R U F2: l2] (11) [y2 x' Lw2 U R U F2: l2] (11)
  [l': [U2, r' D2 r]](10)	  [l2: [r' D2 r, U]](10)	  [l2: [U', r' D2 r]](10)

Case 2: The Right Wing Is Solved

Similar to Case 1, we have 6 cases and can handle each of them with several algorithms.

		
y' + Case	y' + Case	y2 + Case
  [r2 U2 F M' U: r2] (11)	  [r2 U F2 U: r2] (9)	  [y L F: [r2 U F2 U: r2]] (13)
  [y' R' F': [l2 U' F2 U': l2]] (13)	  [L' B': [r2 U F2 U: r2]] (13)	  [y' R' F': [r2 U' F2 U': r2]] (13)
  [R B: [r2 U' F2 U': r2]] (13)	  [F2 U l' r' U F2 U: r2] (15) [B' l2 R' U R F2 U: l2] (15) [r2 U F' R F' U: r2] (13)	  [F2 U l' r' U F2 U: r2] (15) [B' l2 R' U R F2 U: l2] (15) [r2 U F' R F' U: r2] (13)
  [x' r2 U' R U' F2: r2] (11) [y2 x Lw2 U' L' U' F2: r2] (11) [y2 x' Rw2 U' L' U' F2: r2] (11)		  [F2 U' l' r' U' F2 U': r2] (15) [B l2 L U' L' F2 U': l2] (15) [B Rw2 U' F2 L U': r2] (13) [B Rw2 U' L F2 U': r2] (13)
  [r2: [U, l D2 l']](10)	  [r2: [l D2 l', U]](10)	  [r: [l D2 l', U2]](10)

		
y2 + Case	y + Case	y + Case
 [l2 U' F2 U': l2] (9)	 [r2 U2 F' M' U': r2] (11)	 [r2 B2 U' l2 U: l2] (11)
 [l2 U F2 U: l2] (9)	 [y L F: [l2 U F2 U: l2]] (13)	 [r2 U' F2 U': r2] (9)
 [x' r2 U' R U' F2: r2] (11) [y2 x Lw2 U' L' U' F2: r2] (11) [y2 x' Rw2 U' L' U' F2: r2] (11)	 [L' B': [r2 U F2 U: r2]] (13)	 [R B: [r2 U' F2 U': r2]] (13)
 [y2 x' r2 U R U F2: r2] (11) [x Lw2 U L' U F2: r2] (11) [x' Rw2 U L' U F2: r2] (11)	 [y2 x' r2 U R U F2: r2] (11) [x Lw2 U L' U F2: r2] (11) [x' Rw2 U L' U F2: r2] (11)	 [F2 U' l' r' U' F2 U': r2] (15) [B l2 L U' L' F2 U': l2] (15) [B Rw2 U' F2 L U': r2] (13) [B Rw2 U' L F2 U': r2] (13)
 [r: [U2, l D2 l']](10)	 [Rw2: [U', l D2 l']](10)	 [Rw2: [l D2 l', U']](10)

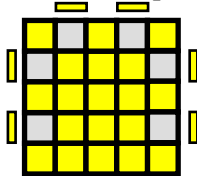
Step 2: OLL

Prerequisite: One Solved dedge.

Assuming that all wings are not oriented correctly

OLLs (19, 12)

Again we can use the 2 2-cycles from step 1. Only one new algorithm is needed to handle its corresponding case. Note that it might be worth it to learn both odd and even permutations algorithms for this case, as move count will be lowered if the cube was in an even permutation to begin with.



Odd Permutation Algorithms

The following algorithm only works on the 4x4x4 cube.

$y R w' U^2 r U^2 R w' x' U^2 r' U' R' U' R w' U^2 R w U R U' R w R U^2 x$ (19)

These three algorithms are directly translatable to all cube sizes, just as the rest are.

$[M' U x: l U^2 r' F^2 r' U^2 l^2 U^2 B^2 r' F^2 r B^2 r^2 U^2]$ (19)

$[M^2 U: r^2 U^2 r U^2 l' U^2 l U^2 r' F^2 r F^2 r' U^2 l' r' U^2 l]$ (22)

$[M^2 U: l r U^2 l U^2 r' U^2 r U^2 r' U^2 l U^2 l U^2 l r U^2 r']$ (23)

Even Permutation Algorithm

$[M^2 U: r D^2 l' B^2 U^2 l^2 D^2 r' U^2 r D^2 l' B^2 D^2 r']$ (19)

The gray colored dedges indicate the dedges which are unsolved.

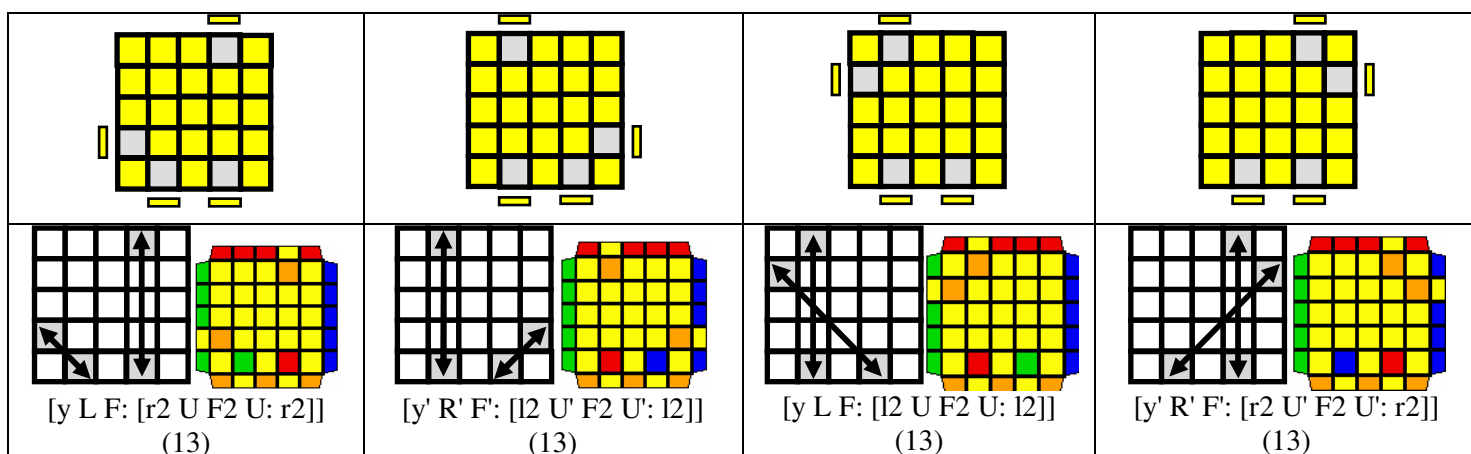
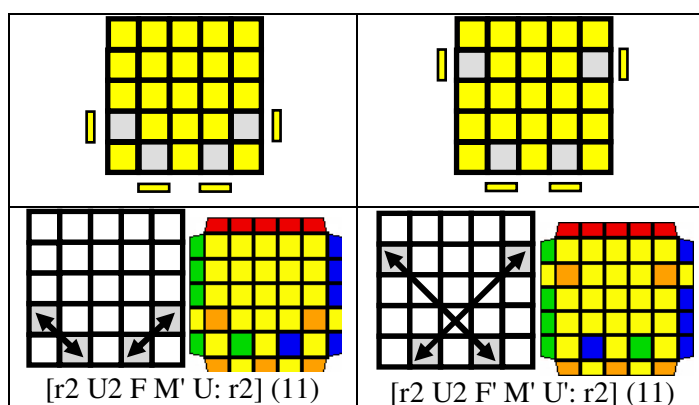
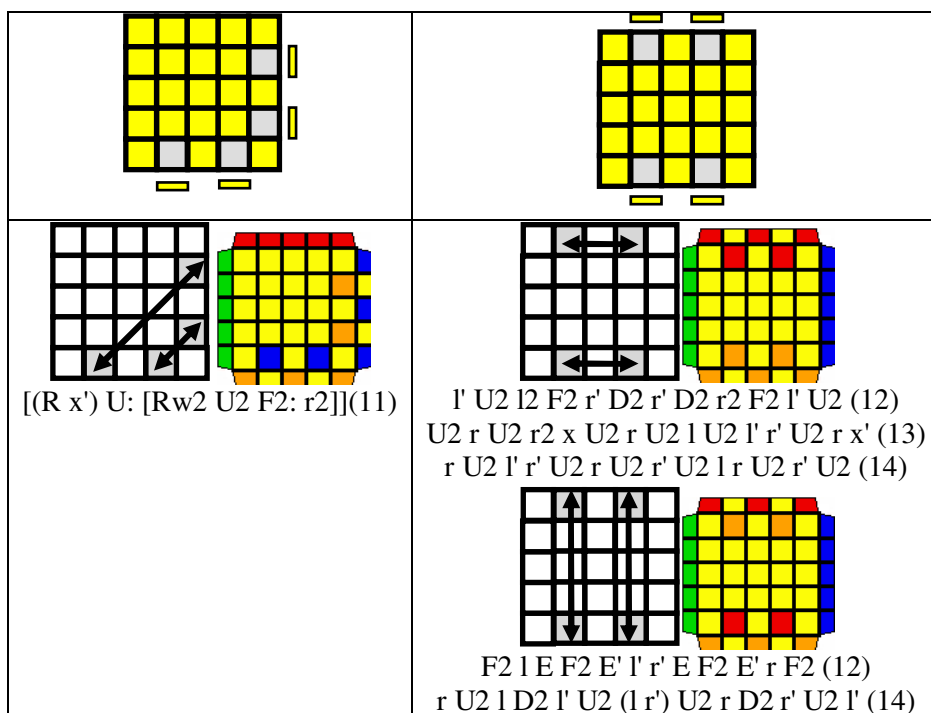
 [l2: [U', r' D2 r]](10)	 [l': [r' D2 r, U2]](10)	 [r2: [U, l D2 l']](10)
 [Rw2: [U', l D2 l']](10)	 [r: [l D2 l', U2]](10)	 [Lw2: [U, r' D2 r]](10)

For the next 6 cases shown, it is ambiguous which dedge was solved in step 1. Hence, I will list the 2 2-cycle algorithms which handle both possibilities for each of the three OLL cases which oriented two wings. The third unsolved dedge is gray.

 [L' B': [l2 U F2 U: l2]] (13)	 [R B: [r2 U' F2 U': r2]] (13)	 [L' B': [r2 U F2 U: r2]] (13)	 [R B: [l2 U' F2 U': l2]] (13)
y' + Case	y + Case	y' + Case	y + Case
 [x' r2 U' R U' F2: r2] (11) [y2 x Lw2 U' L' U' F2: r2] (11) [y2 x' Rw2 U' L' U' F2: r2] (11)	 [x' l2 U L' U F2: l2] (11) [y2 x Rw2 U R U F2: l2] (11) [y2 x' Lw2 U R U F2: l2] (11)	 [y2 x' l2 U' L' U' F2: l2] (11) [x Rw2 U' R U' F2: l2] (11) [x' Lw2 U' R U' F2: l2] (11)	 [y2 x' r2 U R U F2: r2] (11) [x Lw2 U L' U F2: r2] (11) [x' Rw2 U L' U F2: r2] (11)

 [F2 U l' r' U F2 U: r2] (15) [B' l2 R' U R F2 U: l2] (15) [r2 U F' R F' U: r2] (13)	 [F2 U l r U F2 U: l2] (15) [B' r2 R' U R F2 U: r2] (15) [B' Lw2 U F2 R' U: l2] (13) [B' Lw2 U R' F2 U: l2] (13)
y2 + Case	y2 + Case
 [F2 U' l r U' F2 U': l2] (15) [B r2 L U' L' F2 U': r2] (15) [l2 U' F L' F U': l2] (13)	 [F2 U' l' r' U' F2 U': r2] (15) [B l2 L U' L' F2 U': l2] (15) [B Rw2 U' F2 L U': r2] (13) [B Rw2 U' L F2 U': r2] (13)

The rest of the cases are straight forward, as they involve only 4 wings (which is what 2 2-cycles involve).

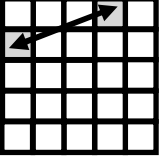
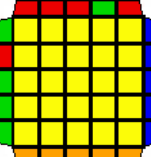
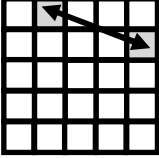
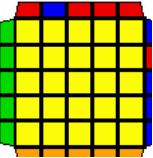


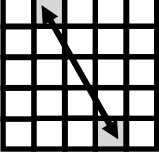
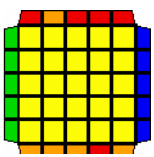
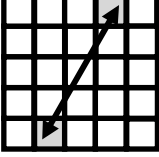
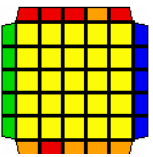
Step 3: PLL

Now 3 or less dedges have pieces to be permuted. Your cube will have one of the 32 cases shown.

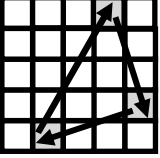
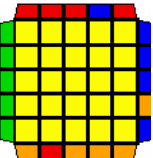
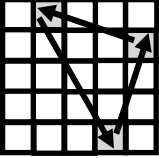
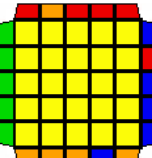
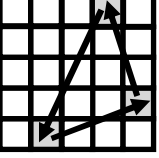
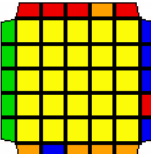
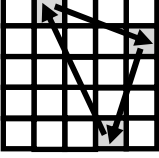
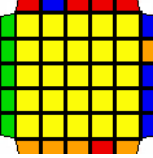
PLLs (32, 14, 7-9)

2 Wings 2-cycles (4, 2, 2)

  $x' l' U' L' U' l' U L U B2 l' B2 r' U2 r' U2 l2 x$ (16) $x' l2 U2 r' U2 l' U' L' U' l' U L U l' r' U2 l' U2 l x$ (18)	  $x' r' U R U r' U' R' U' B2 r' B2 l' U2 l' U2 r2 x$ (16) $x' r2 U2 l' U2 r' U R U r' U' R' U' l' r' U2 r' U2 r' x$ (18)
--	---

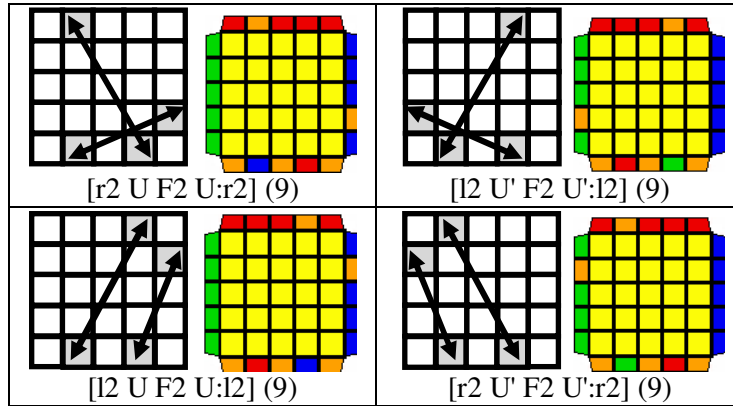
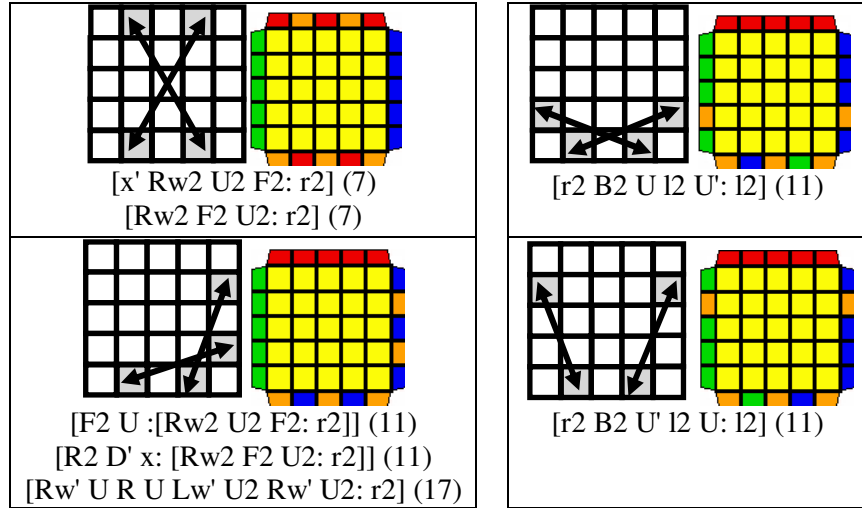
  $r2 F2 U2 r' U2 r2 F2 r' U2 r2 U2 F2 r' F2$ (14) $l' U2 l' U2 l' U2 r' U2 l' U2 l' U2 F2 l2 F2 r$ (16) $Rw' U2 r' U2 r' U2 l' U2 r' U2 r' l' U2 r2 U2 Rw$ (18)	  $l2 F2 U2 l' U2 l2 F2 l' U2 l2 U2 F2 l' F2$ (14) $r' U2 r' U2 r' U2 l' U2 r' U2 r' U2 F2 r2 F2 l'$ (16) $Lw' U2 l' U2 l' U2 r' U2 l' U2 l' U2 r' l' U2 l2 U2 Lw'$ (18)
--	--

3 Wings 3-cycles (4, 1, 1)

  $Rw U' R' U' r' U R U' r' U2$ Rw' (11) $F2 Rw2 U R2 U' r2 U R2 U'$ $R2 F2$ (11)	  $Rw' U R U r' U' R' U r' U2$ Rw (11) $B2 Rw2 U' R2 U r2 U' R2 U$ $R2 B2$ (11)	  $Rw U2 r' U R' U' r' U R U$ Rw' (11) $F2 R2 U R2 U' r2 U R2 U'$ $Rw2 F2$ (11)	  $Rw' U2 r' U R U r' U' R' U'$ Rw (11) $B2 R2 U' R2 U r2 U' R2 U$ $Rw2 B2$ (11)
---	---	---	--

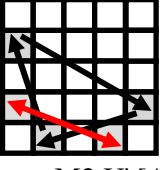
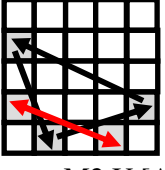
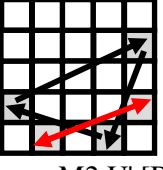
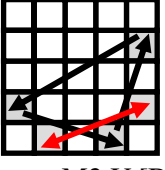
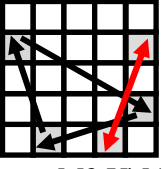
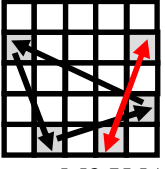
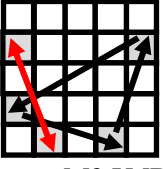
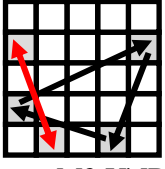
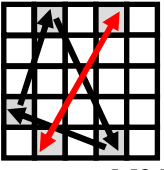
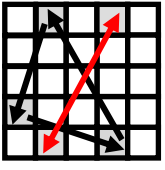
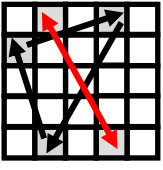
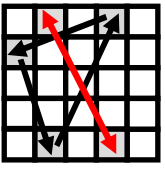
4 Wings

2 2-cycles (8, 6, 3)



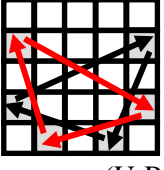
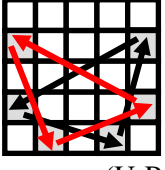
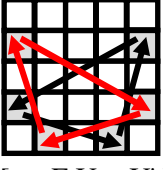
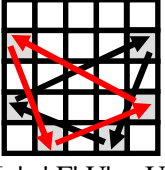
5 Wings

1 2-cycle and 1 3-cycle (12, 3, 1)

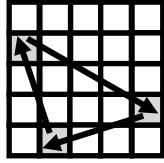
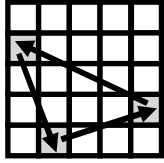
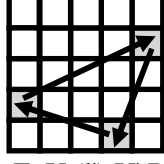
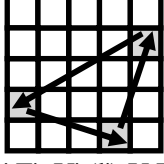
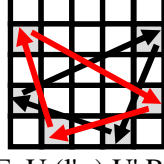
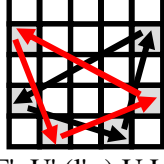
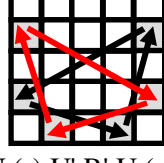
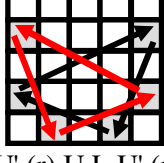
$1\ U2\ x'\ Lw2\ U2\ 1\ U2\ Lw2\ U2\ F2\ l'\ F2\ U2\ l2\ x = [A]$ $x'\ l2\ U2\ F2\ l\ F2\ U2\ Lw2\ U2\ l'\ U2\ Lw2\ x\ U2\ l' = [A]'$ $r'\ U2\ x'\ Rw2\ U2\ r'\ U2\ Rw2\ U2\ F2\ r\ F2\ U2\ r2\ x = \text{Mirror of } [A] = [B]$ $x'\ r2\ U2\ F2\ r'\ F2\ U2\ Rw2\ U2\ r\ U2\ Rw2\ x\ U2\ r = [B]'$			
 M2 U' [A]' U' M2 (17)	 M2 U [A] U M2 (17)	 M2 U' [B] U' M2 (17)	 M2 U [B]' U M2 (17)
 M2 U' [A] U' M2 (17)	 M2 U [A]' U M2 (17)	 M2 U [B] U M2 (17)	 M2 U' [B]' U' M2 (17)
 M2 U x' U2 [A]' x U' M2 (18)	 M2 U x' [A] U2 x U' M2 (18)	 M2 U x' U2 [B]' x U' M2 (18)	 M2 U x' [B] U2 x U' M2 (18)

6 Wings

2 3-cycles (4, 2, 2)

 (U-Perm) F2 U l' r U2 l r' U F2 (9)	 (U-Perm) F2 U l' r U2 l r' U' F2 (9)	 [y z F U: r U' R' U r' l' U' R U l] (14) Rw2 U2 R2 U r2 U' Rw2 U' r2 U R2 U' r2 U Rw2 U r2 U Rw2 (19)	 [y' z' F' U': r U L U' r' l' U L' U' l] (14) Rw2 U' r2 U' Rw2 U' r2 U R2 U' r2 U Rw2 U r2 U' R2 U2 Rw2 (19)
---	--	---	---

Note that you can use variations of a Generic Algorithm to handle all 3-cycle and 2 3-cycle cases. Therefore, one may even consider all 8 cases to be handled with only one algorithm instead of 3.

 $[y\ z\ F: U\ (r)\ U'\ R'\ U\ (r')\ U'\ R]\ (10)$	 $[y'\ z'\ F': U'\ (r)\ U\ L\ U'\ (r')\ U\ L']\ (10)$
 $[y\ z\ F: U\ (l')\ U'\ R'\ U\ (l)\ U'\ R]\ (10)$	 $[y'\ z'\ F': U'\ (l')\ U\ L\ U'\ (l)\ U\ L']\ (10)$
 $[y\ z\ F: U\ (l'\ r)\ U'\ R'\ U\ (r'\ l)\ U'\ R]\ (12)$	 $[y'\ z'\ F': U'\ (l'\ r)\ U\ L\ U'\ (r'\ l)\ U\ L']\ (12)$
 $[y\ z\ F: U\ (r)\ U'\ R'\ U\ (r'\ l')\ U'\ R\ U\ (l)\ U']\ (14)$	 $[y'\ z'\ F': U'\ (r)\ U\ L\ U'\ (r'\ l')\ U\ L'\ U'\ (l)\ U]\ (14)$