

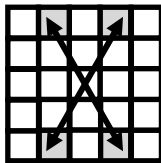
A Complete Set of Brief Algorithms for all 58 *Two 2-Cycle* Last Layer Wing Edge Cases

By: Christopher Mowla

Notes

- In this document, I provide at least one algorithm for each of the 58 2 2-cycle cases; I listed the algorithms with lowest btm (block (half) turn moves) first. Every case has a pure (supercube safe) solution under it.
- Sometimes there is a '[4]' to the *left* of the first algorithms listed. This means that this algorithm only works on the 4x4x4 cube.
- All algorithms without a '[4]' to the left of them should be translatable to all size cubes. With that being said, note that the moves M and E mean all inner layer slices, not just the central slice on odd cubes. I did not count (l' r) as one move because it is two moves on every cube size except the 4x4x4. The move M is one move on all size cubes because it is, as I have said before, telling you to move ALL inner layer slices.
- I am not claiming that any of the algorithms I have listed under each permutation diagram are the optimal solutions for any given move metric, but I believe most of lowest move algorithms provided for each case are optimal in btm.
- All algorithms are in WCA notation.
- For those who don't know, an algorithm of the form [A: B] = A B A' ('A' here is called a conjugate). In addition, an algorithm of the form [A, B] = A B A' B' (this is called a commutator).
- After some of the algorithms, I have written the person who I believe should be credited for the algorithm. If there are more independent founders, I apologize. Therefore, all algorithms which have no name following them were found by me by hand. Many of the algorithms which I attribute credit to others were not (at least not to my knowledge) created by them entirely, but I made those algorithms from the algorithms which they did make.
- This document is by no means a guarantee to speed up your K4LL solving time, but there could be a few algorithms you might like. As Thom Barlow mentioned on his ELL page (Step 7), "...you can sometimes solve a 2x2-cycle faster by using 2 3-cycles." I definitely agree for cases which require more moves, but some of the algorithms in this document are worth memorizing.
- Some of the shortest algorithms listed are definitely not as fast to execute as the longer algorithm following them, but I included these brief solutions for fun.
- A huge portion of the algorithms I created are just conjugates of l2 or r2. In my opinion, these algorithms are even easier to understand than commutators which can solve the same problem. The only problem with these algorithms is that they are not pure (not supercube safe). In other words, they are often briefer than pure algorithms, but they swap same color centers with each other, which is not useful when solving supercubes. I label the supercube safe algorithms as pure.
- This collection of algorithms is by no means complete in any way as far as the "best" algorithms that might exist, whether referring to having a low move count or being speed optimal. There are probably many algorithms to be found by various methods, and it's very possible that briefer (and possibly faster) solutions exist already which I am not aware of.
- I know that a 2-gen 4x4x4 last layer solver was created and made available on IRC, but I am assuming that those solutions are more plain turn metric moves (ftm and qtm) than the algorithms in this document. I did not restrain myself to a particular move set when making these algorithms, so they are most likely not longer in btm than the 2-gen algorithms, especially 2-gen algorithms that work for all size cubes (usually, the stricter the move set, the longer the algorithms are). As you will see, I did make an effort nonetheless to leave out very ugly solutions such as [4] [(L2 x2) u2 R2 (U y') M: b2] (11), even though it is the same number of moves as the briefest solution which I did choose to include which is: [4] [Uw2 r2 U l2 U': l2] (11). However, for the most part, I omitted longer algorithms.
- I used 5x5x5 permutation images because it presents the permutation more clearly (in my opinion).
- If you are unsure how to execute a particular algorithm, just copy and paste it into alg.garron.us, select the 4x4x4 cube, select 'WCA notation', and click 'view'. As I said before, if you need to execute an algorithm on a bigger cube, take the 5x5x5 for example, then M = (l M r'), M' = (l' M' r), M2 = (l2 M2 r2), E = (u' E d), E' = (u E' d'), S = (f S b'), and S' = (f' S' b).

One of the sources credited is "AF" (Algorithm Finder) followed by :S_-. Those algorithms are from the "seed algorithms" on this page (http://www.mementoslangues.fr/CubeDesign/CubeAlgorithms/DataBase_TwoTwoCycles_Edges.txt). The number following the S is the seed algorithm number. Note that several algorithms that existed prior to generation date can be made from these seed algorithms, but I did not credit that source for algorithms which I definitely knew existed before.



[4]r2 U2 r2 Uw2 r2 u2 (6)

(Chris Hardwick)

[4] [Uw2 Rw2 U2: r2] (7)

(Stefan Pochmann)

[Rw2 F2 U2: r2] (7)

(Stefan Pochmann)

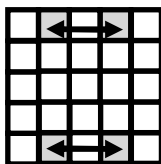
r2 U2 B2 l r U2 (l' r) U2 r2 B2 U2 (12) [Pure]

(Stefan Pochmann)

r2 U2 r U2 r2 U2 r2 U2 r U2 r2 U2 (12) [Pure] (?)

r' F U' R F' U l r U' F R' U F' l' (14) [Pure]

(From standard edge pairing alg.)



l' U2 l2 F2 r' D2 r' D2 r2 F2 l' U2 (12) [Pure]

(Cube Explorer)

U2 r U2 r2 x U2 r U2 l U2 l' r' U2 r x' (13) [Pure]

(Cube Explorer)

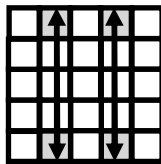
F2 r E F2 E' l r E F2 E' l' r2 F2 (13) [Pure]

r U2 l' r' U2 r U2 r' U2 l r U2 r' U2 (14) [Pure]

(Kenneth Gustavsson)

(l r) U' F R' U F' (l' r) F U' R F' U (14) [Pure]

(From standard edge pairing alg.)

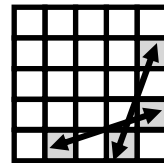


F2 l E F2 E' l' r' E F2 E' r F2 (12) [Pure]

F2 l E' F2 E l' r' E' F2 E r F2 (12) [Pure]

r U2 l D2 l' U2 (l r) U2 r D2 r' U2 l' (14) [Pure]

(Kenneth Gustavsson)



[4][R2 D' x: r2 U2 r2 Uw2 r2 u2] (10)

(Chris Hardwick)

[F2 U : [Rw2 U2 F2: r2]] (11)

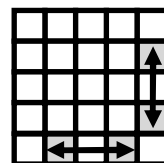
(Stefan Pochmann)

[R2 D' x: [Rw2 F2 U2: r2]] (11)

(Stefan Pochmann)

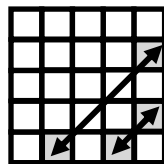
[y' R' F: l E F2 E' l' r' E F2 E' r] (14) [Pure]

[Rw' U R U Lw' U2 Rw' U2: r2] (17)



F' R' U F' (l' r) F U' R F' U (l r) U' F2 (15) [Pure]

(From standard edge pairing alg.)



[4][(R x') U x: r2 U2 r2 Uw2 r2 u2] (10)

(Chris Hardwick)

[(R x') U: [Rw2 U2 F2: r2]] (11)

(Stefan Pochmann)

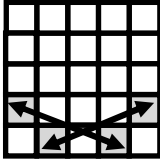
[(Lw r') U Lw2 U2 B2: r2] (11)

(Stefan Pochmann)

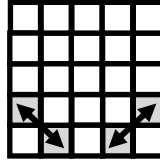
R B U2 r2 U2 B2 l r U2 (l' r) U2 r2 B R' (15) [Pure]

(Stefan Pochmann)

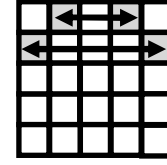
Symmetric Cases



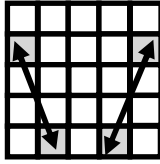
[r2 B2 U l2 U': l2] (11)
[r': [L F' L', D' f' D]] (14) [Pure]
(AF: S1)



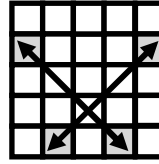
[r2 U2 F M' U: r2] (11)
[L' F u2 F' L, U] (12) [Pure]
(Thom Barlow)



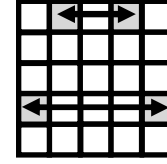
[x l2 U F L U R2 F: r2] (15)
[l R2 y: [r D2 r', l' U' l]] (16) [Pure]
(Thom Barlow)



[r2 B2 U' l2 U: l2] (11)
l D' b D L F' L' D' b' D L F Lw' (13) [Pure]
(AF: S2)

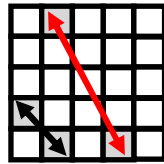


[r2 U2 F' M' U': r2] (11)
[L' F d2 F' L, U] (12) [Pure]
(Thom Barlow)

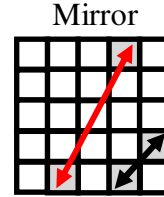


[x r2 U F L U R2 F: l2] (15)
[l L2 y': [r D2 r', l' U' l]] (16) [Pure]
(Thom Barlow)

Diagonal swap between two opposite composite edges (red)

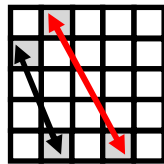


[Rw2 F2 U2 x' U' R U: r2] (13)
[Rw2 y': [L2, U' r' (E y) r2 D]] (14) [Pure]
(AF: S60)

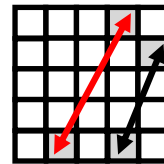


Mirror

[Lw2 F2 U2 x' U' L' U': l2] (13)
[Lw2 y: [R2, U l (E' y') l2 D']] (14) [Pure]
(AF: S60)

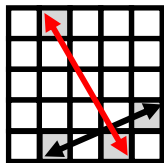


[Rw2 U' F2 U': r2] (9)
[Rw2 F2 L' U L U: r2] (13)
[B2: [r2, D F' L F D']] (14) [Pure]
(AF: S16)

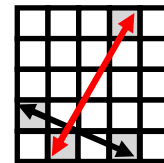


Mirror

[Lw2 U F2 U: l2] (9)
[Lw2 F2 R U' R' U': l2] (13)
[B2: [l2, D' F R' F' D]] (14) [Pure]
(AF: S16)

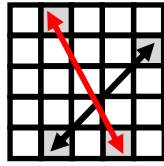


[r2 U F2 U: r2] (9)
[Lw2 F2 R' U R U: l2] (13)
[B2: [r2, D' F R' F' D]] (14) [Pure]
(AF: S15)

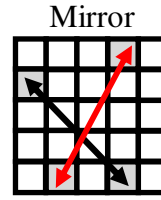


Mirror

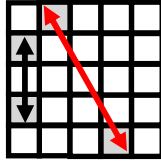
[l2 U' F2 U': l2] (9)
[Rw2 F2 L U' L' U': r2] (13)
[B2: [l2, D F' L F D']] (14) [Pure]
(AF: S15)



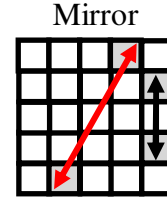
[x Rw2 U' R U' F2: l2] (11)
[r2: [B2, (D y) r2 E b' U']] (14) [Pure]
(AF: S54)



Mirror
[x Lw2 U L' U F2: r2] (11)
[l2: [B2, (D' y') l2 E' b U]]
(AF: S54)



[4] [y' l': [U', l' r' U2 l r]] (13)
(Thom Barlow)



Mirror
[4] [y r: [U, l r U2 l' r']] (13)
(Thom Barlow)

[4][f: Lw2 F2 U2 l2 U2 Lw2 U' Lw2 F2 Lw2 U'] (13)

[4][f: Rw2 F2 U2 r2 U2 Rw2 U Rw2 F2 Rw2 U] (13)

[y r': [U, l' r' D2 l r]] (13) [Pure]

[y' l: [U', l r D2 l' r']] (13) [Pure]

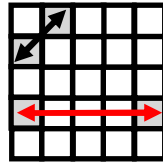
(Thom Barlow)

(Thom Barlow)

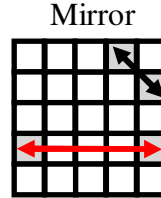
[x' l' U L' U' Lw' U2 F2: l2] (15)

[x' r U' R U Rw U2 F2: r2] (15)

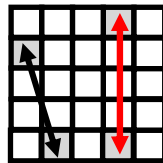
Swap directly across two composite edges (red)



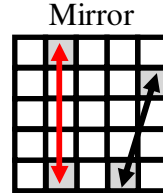
[L F: [r2 U F2 U: r2]] (13)
[Bw2: [U' r U, B R' B']] (14) [Pure]
(AF: S1)



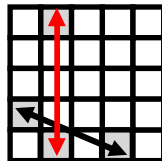
Mirror
[R' F': [l2 U' F2 U': l2]] (13)
[Bw2: [U l' U', B' L B]] (14) [Pure]
(AF: S1)



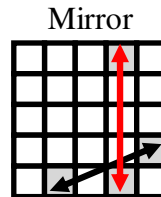
[B Rw2 U' F2 L U': r2] (13)
[B Rw2 U' L F2 U': r2] (13)
[Rw: [L B u' B' L', U]] (14) [Pure]
(AF: S1)
[F2 U' l' r' U' F2 U': r2] (15)



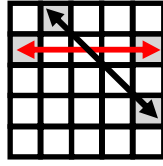
Mirror
[B' Lw2 U F2 R' U: l2] (13)
[B' Lw2 U R' F2 U: l2] (13)
[Lw': [R' B' u B R, U']] (14) [Pure]
(AF: S1)
[F2 U l r U F2 U: l2] (15)



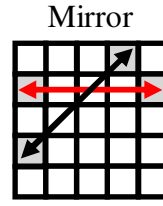
[l2 U' F L' F U': l2] (13)
[l': [U, L B d B' L']] [Pure]
(AF: S2)



Mirror
[r2 U F' R F' U: r2] (13)
[r: [U', R' B' d' B R]] (14) [Pure]
(AF: S2)

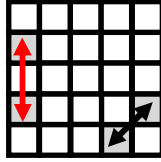


[R' F': [Rw2 U' F2 U': r2]] (13)
 [b2: [B' R B, D l D']] (14) [Pure]
 (AF: S2)

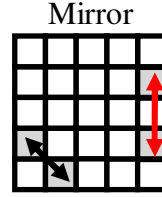


Mirror
 [L F: [Lw2 U F2 U: l2]] (13)
 [b2: [B L' B', D' r' D]] (14) [Pure]
 (AF: S2)

“Single edge flip” without either of the previously two red colored 2-cycle arrows



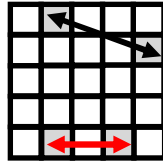
[Bw: [L2, U' r' (E y) r2 D]] (14) [Pure]
 (AF: S60)



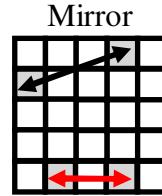
Mirror
 [Bw': [R2, U l (E' y') l2 D']] (14) [Pure]
 (AF: S60)

F2 U' l' r' U F' L U' F l r F' U L' F' (15) [Pure]
 (From standard edge pairing alg.)

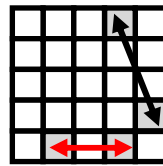
F2 U l r U' F R' U F' l' r' F U' R F (15) [Pure]
 (From standard edge pairing alg.)



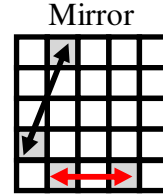
[r2 U' B2 U' x U L U': r2] (15)
 [y r': F' L U' F l r F' U L' F U' l' r' U] (16) [Pure]
 (From standard edge pairing alg.)



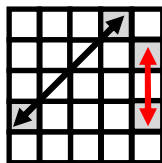
Mirror
 [l2 U B2 U x U' R' U: l2] (15)
 [y' l: F R' U F' l' r' F U' R F' U l r U'] (16) [Pure]
 (From standard edge pairing alg.)



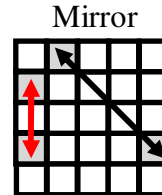
[Lw y': [R2, D r (E y) r' U]] (14) [Pure]
 (AF: S26)
 [Lw2 U' B2 U' F L F': l2] (15)



Mirror
 [Rw' y: [L2, D' l' (E' y') l U]] (14) [Pure]
 (AF: S26)
 [Rw2 U B2 U F' R' F: r2] (15)



[b': [R2, D' r2 (y' E') r U]] (14) [Pure]
 (AF: S54)

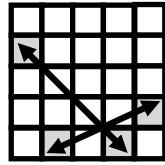


Mirror
 [b: [L2, D l2 (y E) l' U]] (14) [Pure]
 (AF: S54)

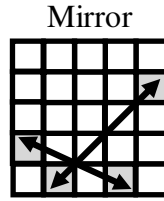
[r2: U l r U' F R' U F' l' r' F U' R F'] (16) [Pure]
 (From standard edge pairing alg.)
 [y2 F2 r' F M' U' Rw' F2 D2: l2] (17)

[l2: U' l' r' U F' L U' F l r F' U L' F] (16) [Pure]
 (From standard edge pairing alg.)
 [y2 F2 l F' M' U Lw F2 D2: r2] (17)

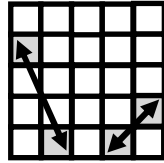
Other



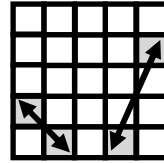
[L' B': [r2 U F2 U: r2]] (13)
[r2: [U, b l D l' b']] (14) [Pure]
(AF: S5)



Mirror
[R B: [l2 U' F2 U': l2]] (13)
[l2: [U', b' r' D' r b]] (14) [Pure]
(AF: S5)

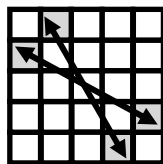


[R B: [r2 U' F2 U': r2]] (13)
[r2: [U', l f D' f l']] (14) [Pure]
(AF: S5)

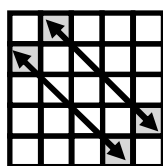


Mirror
[L' B': [l2 U F2 U: l2]] (13)
[l2: [U, r' f' D f r]] (14) [Pure]
(AF: S5)

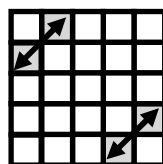
Symmetric Cases



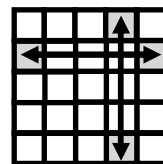
[Rw2 U' R2 F2 U': r2] (11)
 [Rw2 U' F2 R2 U': r2] (11)
 [R' Rw2 U' F2 R' U': r2] (13)
 [L' r2 U F2 L U: r2] (13)
 [x' Rw2: [u2, R' L' U L R]] (14) [Pure]
 (AF: S19)



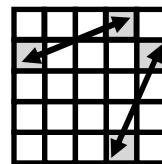
[L' F d2 F' L, U2] (12) [Pure]
 (Thom Barlow)
 [M2 U: r' U2 r D2 r' U2 l' r U2 l D2 l' U2 l] (18)
 [Pure]
 (Kenneth Gustavsson)



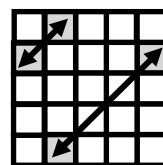
[F' R u2 R' F, U2] (12) [Pure]
 (Thom Barlow)
 [y l r: (D' y') F2 E' l r E F2 E' l' r' U] (15) [Pure]
 [Bw2: U' r D' R2 D r' D' U r U' R2 U r' D] (16)
 [Pure]



[x R2: r' U' L R U r' U' R' U r2 U' L' U] (15) [Pure]
 (AF: S?)

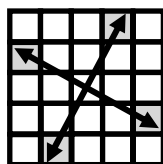


[4][Uw2 r2 U l2 U': l2] (11)
 r' (E y) r U' L2 U r' (E' y') r' D' L2 D r2 (13) [Pure]
 (AF: S28)
 [r': F' R' U F r' F' U' R F U r U'] (14) [Pure]
 (AF: S7)
 r2 D' R2 D U r' U' D' R2 U D r U' r2 (14) [Pure]
 r2 U' (r2 U2)2 r' (U2 r2)2 U2 r' U' r2 (15) [Pure]

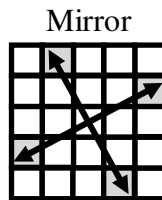


[r2 F2 U2 B M U': r2] (13)
 [y: r D' R2 D r (E y) r' U' R2 U r (E' y') r2] (13)
 [Pure]
 (AF: S26)

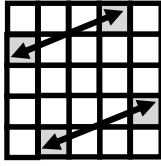
Other



[4] (Rw2 B2 Rw2' U)2 (8)
 (Thom Barlow)
 [l r U2 l' r', U] (12)
 (Thom Barlow)
 [U', L R u2 R' L'] (12) [Pure]
 (Per Kristen Fredlund)
 [l r D2 l' r', U] (12) [Pure]
 (Thom Barlow)



Mirror
 [4] (Lw2 B2 Lw2' U)2 (8)
 (Thom Barlow)
 [l' r' U2 l r, U'] (12)
 (Thom Barlow)
 [U, R' L' u2 L R] (12) [Pure]
 (Per Kristen Fredlund)
 [l' r' D2 l r, U'] (12) [Pure]
 (Thom Barlow)



[y': L U' R u2 R' U M x U' L u2 L' U R'] (13) [Pure]

(AF: S30)

M2 U' l' r' E F2 E' l r E F2 (D y) M2 (13) [Pure]

[R: F L u L' F' U' F L u' L' F' U] (14) [Pure]

(AF: S1)

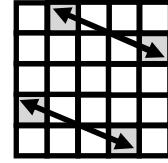
[Bw2: U l2 D L2 D' l2 D U' l2 U L2 U' l2 D'] (15)
[Pure]

[l U' r D2 r' U l', U2] (16) [Pure]

(Thom Barlow)

[y x U2 l2 U F' L U R2 F': r2] (17)

Mirror



[y: R' U L' u2 L U' M x U R' u2 R U' L] (13) [Pure]

(AF: S30)

M2 U l r E' F2 E l' r' E' F2 (D' y') M2 (13) [Pure]

[L': F' R' u' R F U F' R' u R F U] (14) [Pure]

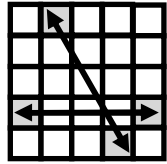
(AF: S1)

[Bw2: U' r2 D' R2 D r2 D' U r2 U' R2 U r2 D] (15)
[Pure]

[r' U l' D2 l U' r, U2] (16) [Pure]

(Thom Barlow)

[y' x U2 r2 U' F R' U' L2 F: l2] (17)



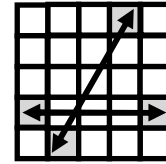
[R2 y: [l' D2 l, r U r']] (14) [Pure]

(Thom Barlow)

[r2 U F2 U x' U' R U: r2] (15)

[y r' U2 M' r: U r' U2 r' U2 r U2 r U] (17)

Mirror

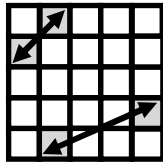


[L2 y': [r D2 r', l' U' l]] (14) [Pure]

(Thom Barlow)

[l2 U' F2 U' x' U L' U': l2] (15)

[y' l U2 M' l': U' l U2 l U2 l' U2 l' U'] (17)

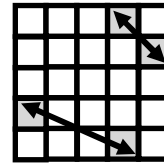


S z' R' d R F U' F' R' d' R F U B' (13) [Pure]

(AF: S2)

[r2 U F2 U x' U R U': r2] (15)

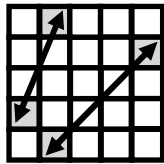
Mirror



S' z L d' L' F' U F L d L' F' U' B (13) [Pure]

(AF: S2)

[l2 U' F2 U' x' U' L' U: l2] (15)



F U' B' L' u' L B U B' L' u L S z' (13) [Pure]

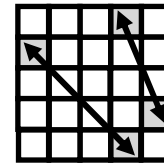
(AF: S1)

[l': [U, R' B' u B R]] (14) [Pure]

(AF: S1)

[r2 U B2 U x U L U': r2] (15)

Mirror



F' U B R u R' B' U' B R u' R' S' z (13) [Pure]

(AF: S1)

[r: [U', L B u' B' L]] (14) [Pure]

(AF: S1)

[l2 U' B2 U' x U' R' U: l2] (15)