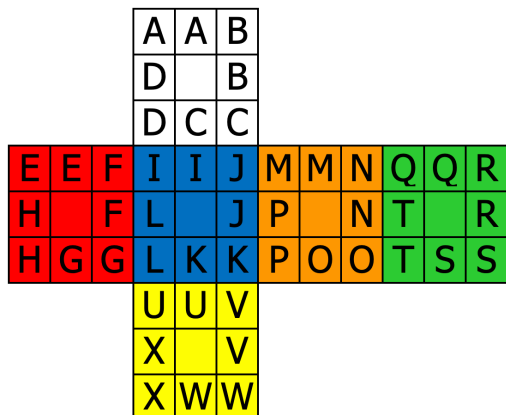


# Blindsolving Cheatsheet

## Speffz lettering scheme



### 3x3 steps

Solve edges	M2 method
Perform parity algorithm?	Uneven number of edge swaps
Solve corners	Old Pochmann method

### 4x4 steps

Solve centers	U2 method
Perform U2 parity algorithm?	Uneven number of center swaps
Solve edges	r2 method
Perform r2 parity algorithm?	Uneven number of wing swaps
Solve corners	Old Pochmann method
Perform 4x4 OP parity?	Uneven number of corner swaps

## 3x3 edges: M2

Buffer piece is located on **U** . Swap with piece located on **A** .

### Algorithms

<i>Non M-slice target</i>	Set-up to <b>A</b> (without disrupting M-slice), perform <b>M2</b> , undo set-up
<b>A</b>	<b>M2</b>
<b>Q</b>	<b>(U B' R U' B) M2 (B' U R' B U')</b>
<b>C</b> (even), <b>W</b> (uneven)	<b>U2 M' U2 M'</b>
<b>W</b> (even), <b>C</b> (uneven)	<b>M U2 M U2</b>
<b>I</b> (even), <b>S</b> (uneven)	<b>(D M' U) R2 (U' M U) R2 (U' D' M2)</b>
<b>S</b> (even), <b>I</b> (uneven)	<b>(M2' D U) R2 (U' M' U) R2 (U' M D')</b>
M2/OP parity	<b>(D' L2 D) M2 (D' L2 D)</b>

## 3x3 corners: OP (Old Pochmann)

Buffer piece is located on **A** . Swap with piece located on **P** . Swap by setting up target piece to **P** (without disrupting **A** or **D** ), performing the swapping algorithm, and undoing set-up.

### Algorithms

Swapping algorithm	<b>(R' F) (R2 U' R' U') (R U R' F') (R U R' U')</b>
--------------------	---

## 4x4 centers: U2

Buffer piece is located on **A** . Swap piece is located on **C** .

### Algorithms

<i>Non U or D face target</i>	Set-up to <b>C</b> (without disrupting rest of U face), perform <b>U2</b> , undo set-up
<i>D face target</i>	Set-up to <b>V</b> , perform <b>(r2 D' r2 D r2) U2 (r2 D' r2 D r2)</b> , undo set-up
<b>B</b> (even), <b>D</b> (uneven)	<b>(r' u r) U (r' u' r) U (r' u r) U2 (r' u' r) U2</b>
<b>D</b> (even), <b>B</b> (uneven)	<b>(r' u r) U' (r' u' r) U' (r' u r) U2 (r' u' r) U2</b>
Parity	<b>U2</b>

## 4x4 wings: r2

The r2 method is mostly similar to the M2 method, with r2 interchanges instead of M2. With the r2 method, there is never a piece 'flipped'; both adjacent edge pieces have a different letter (therefore, K is also a specific piece to solve, instead of the buffer flipped).

Each face only names 4 wings: the ones most-clockwise in the edge-pair (there is a gap in between a corner and the edge with the same label). The other edges get their labels via the other faces.

Buffer is piece located on U. Target is piece located on A.

### Algorithms

Non l or r slice piece	Set-up to A (using same set-up as with M2), perform r2, undo set-up
Q	perform (U B' R U' B) r2 (B' U R' B U')
C, K, W	Set-up with l rotation to Q, solve as Q, undo set-up
I (even), S (uneven)	(D r' U) R2 (U' r' U) R2 (U' D' r2)
S (even), I (uneven)	(r2' D U) R2 (U' r U) R2 (U' r D')
Parity	r2 (D' L' F) (l' U2 l' U2) (F2 l' F2 r) (U2 r' U2 l2) (F' L D)

Parity algorithm is based on the following algorithm to swap the UBR and UFR wings: (l' U2 l' U2) (F2 l' F2 r) (U2 r' U2 l2). First a r2 move is done to fix all pieces, except for the U and A pieces. Then the U piece is brought to the UFR location, the swapping is performed, and the U piece set-up is undone.

## 4x4 corners: OP (Old Pochmann)

4x4 corners is the same as for a 3x3, but the parity algorithm is different. Now the parity only swaps the wing-pairs that are equivalent to the 3x3 A and D pieces.

### Algorithms

Parity	(L2 D F2) (r2 U2 r2 Uw2 r2 Uw2) (F2 D' L2)
--------	--

This parity algorithm first sets up both 'edges' to the UF and UB spots, performs the standard 4x4 PLL parity algorithm, and undoes the set-up.

## 4x4 tips

### Centers

- Skip U-face pieces when possible, to limit cycle breaks.
- Shoot other pieces to their face in alphabetical order.

### Wings

- Touch finger when single piece on 'edge' memorized, remove when both memorized.