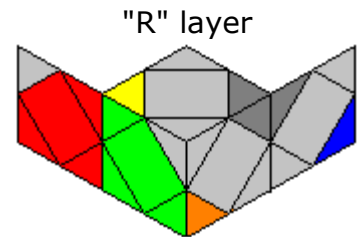
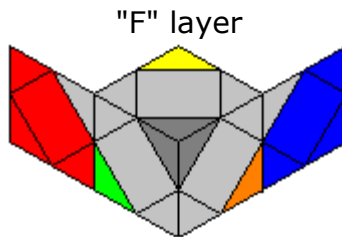
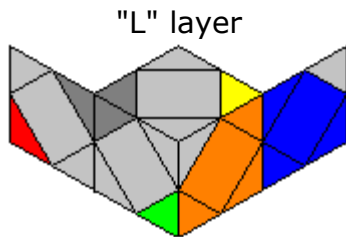


Speedskewbin

Part 1 - Notation and Algs

For simplicity, I'll be using this notation:



This notation is NOT related to FCN. I'll be using this instead of FCN because it's much more convenient to describe the algorithms I use. The x, y, and z rotations will be the same rotations as used in FCN.

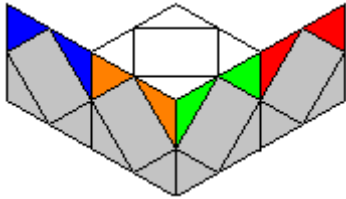
My method is based off of two very easy algorithms. They're both four moves, and they are the same as the "sledgehammer" ($R' F R F'$) and the "hedgeslammer" ($F R' F' R$) on a 3x3x3 cube.

I recommend being familiar with these two algorithms, since you're going to be using them a lot. There are lots of ways to grip the skewb to perform them, there isn't a single best way to do it. Also, figure out for yourself whether you prefer the sledgehammer or hedgeslammer, since you can often control which one you do more often.

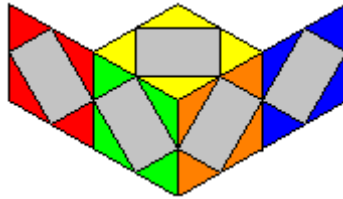
Part 2 - The "Beginners' Variation"

These are the four steps:

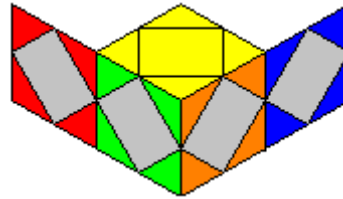
Step 1 –
Solve a side



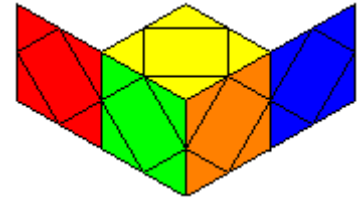
Step 2 –
Solve the remaining
corners



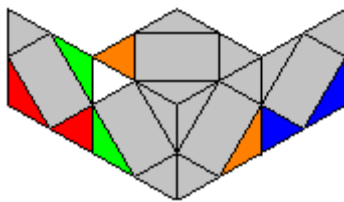
Step 3 –
Solve the U center



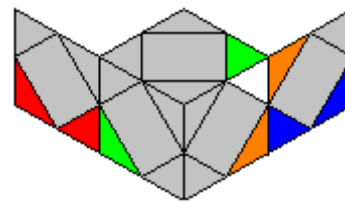
Step 4 –
Solve the remaining
centers



The first step is intuitive. Once you become familiar with how the skewb turns, this is an easy step. When you're trying to solve a side for the first few times, the last corner might be a bit tricky. Two-thirds of the time, you will have to take out an already-solved corner in order to solve the last corner. If this is the case, place the corner so that it is in one of the two positions below, and perform the algorithm below it.

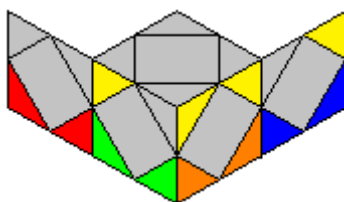


L' F' L'



R' F R

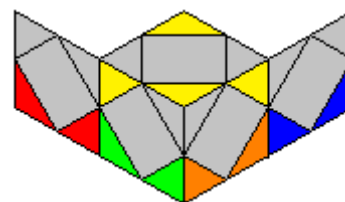
The second step is where the 4-movers come in. If the upper corners aren't already solved, you will have one of these two cases.



F' L F L'

or

(y2) L' F' L' F

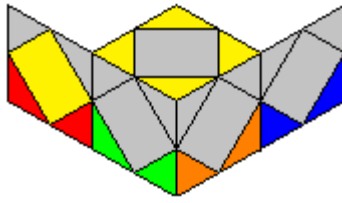


R' F R F' (y) R' F R F'

or

L F' L' F (y') L F' L' F

The third step is simple enough.

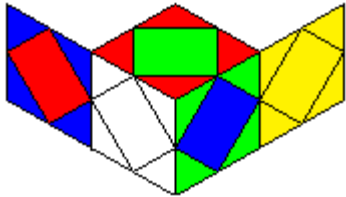


R' F R F' (y2) R' F R F'

or

F R' F' R (y2) F R' F' R

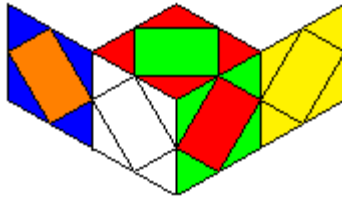
The last step involves the same algorithms used in step 3. Do a (z) or (z') rotation first, to match one of the cases below.



R' F R F' (y2) R' F R F'

or

F R' F' R (y2) F R' F' R



R' F R F' (y2) R' F R F'

(y2 x')

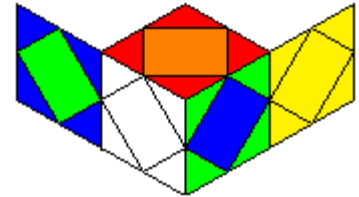
R' F R F' (y2) R' F R F'

or

F R' F' R (y2) F R' F' R

(y2 x')

F R' F' R (y2) F R' F' R



R' F R F' (y2) R' F R F'

(z2 x')

R' F R F' (y2) R' F R F'

or

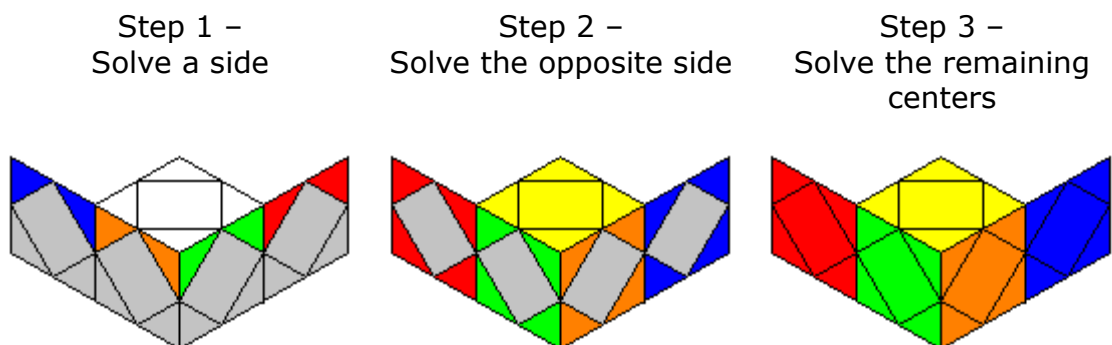
F R' F' R (y2) F R' F' R

(z2 x')

F R' F' R (y2) F R' F' R

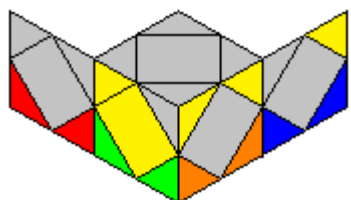
Part 3 - The "Intermediate Variation"

This variation is just like the Beginners' variation, except the second and third steps are combined. These are the steps:

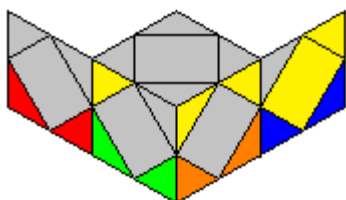


The first and third steps are done the same way as in the Beginner's variation. There are ten cases for the second step, divided into three categories. The category 2 and 3 cases are eventually reduced to category 1 cases.

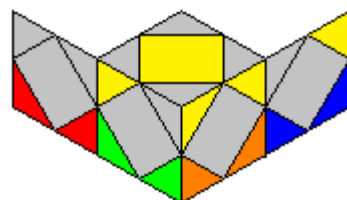
Category 1 Cases



F' L F L'



(y2) L F' L' F

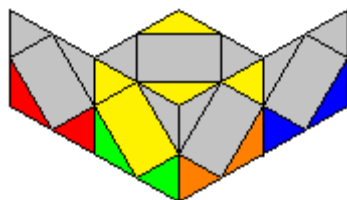


(y2) (F' L F L')2

or

(L F' L' F)2

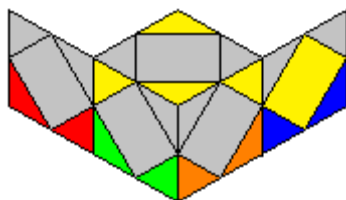
Category 2 Cases



R' F R F' → cat. 1

or

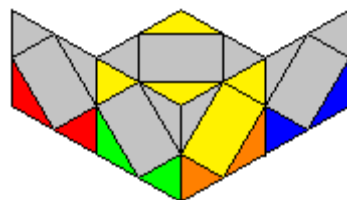
(y2) R' F R F' → cat. 1



R' F R F' → cat. 1

or

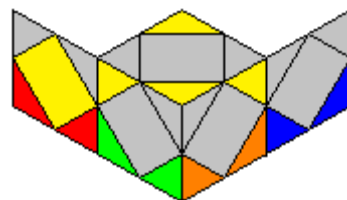
(y2) F R' F' R → cat. 1



L F' L' F → cat. 1

or

(y2) L F' L' F → cat. 1

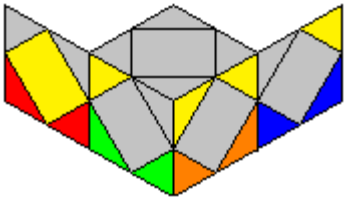


L F' L' F → cat. 1

or

(y2) F' L F L' → cat. 1

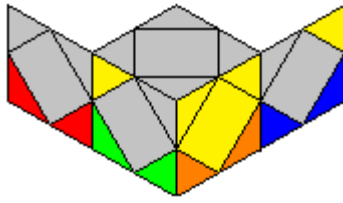
Category 3 Cases



$R' F R F' \rightarrow \text{cat. 2}$

or

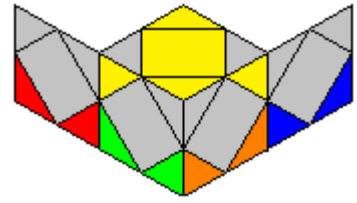
$F R' F' R \rightarrow \text{cat. 2}$



$(y2) R' F R F' \rightarrow \text{cat. 2}$

or

$(y2) F R' F' R \rightarrow \text{cat. 2}$



$R' F R F' \rightarrow \text{cat. 2}$

or

$(y) R' F R F' \rightarrow \text{cat. 2}$

or

$(y') F R' F' R \rightarrow \text{cat. 2}$

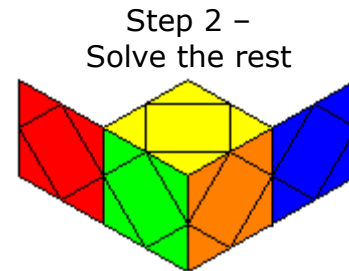
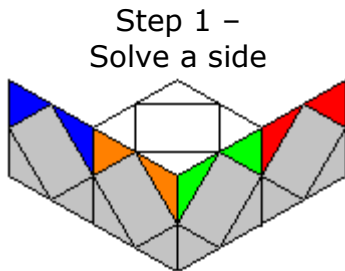
or

$(y2) F R' F' R \rightarrow \text{cat. 2}$

Part 4 - The "Advanced Variation"

This is where it gets a bit crazy. The second and third steps of the intermediate variation are combined into one step.

These are the two steps:



There are 134 cases for the second step. I've organized them into groups based on their algorithms. To view them based on the "CLL + U center" cases, see <http://sarah.cubing.net/skewb/advanced-variation-by-case.pdf>.

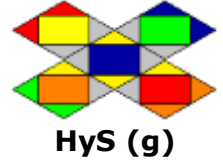
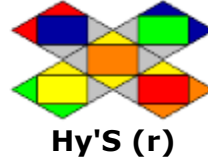
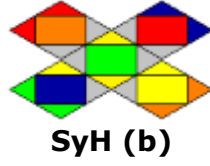
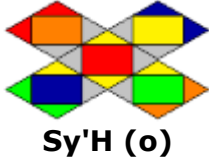
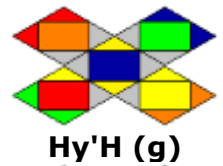
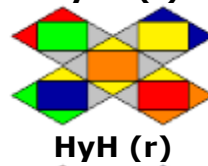
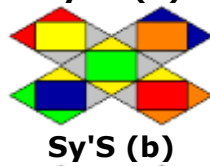
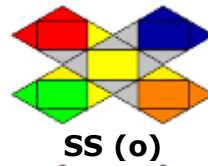
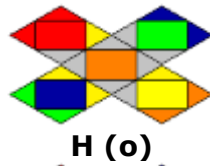
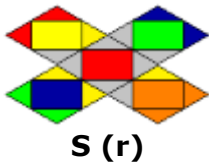
The sledgehammer and hedgeslammer are named S and H respectively, for simplicity, where the sledgehammer is $F' L F L'$ and the hedgeslammer is $L F' L' F$.

The letter in brackets at the end of algorithm indicate which face is at the Front-Left. Green is (g), orange is (o), blue is (b), and red is (r). In other words, (g) indicates no rotation, (o) indicates a (y) rotation, (b) indicates a (y²) rotation, and (r) indicates a (y') rotation.

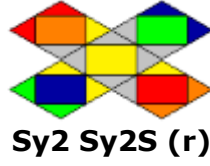
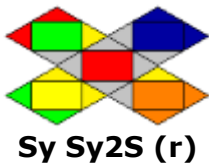
Cases that take 5 sledgehammers/hedgeslammers are marked with with "-", indicating that if you intend to learn these cases, you should probably just learn optimal algs them.

For the most of the Last 5 Centers cases, also marked with "-", it's not very efficient solving them with sledgehammers and hedgeslammers. For optimal algorithms, check out <http://meep.cubing.net/l5c.htm> (note: they're written in FCN).

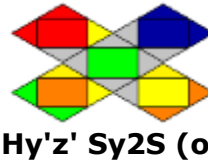
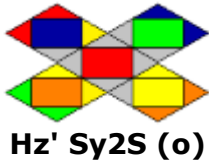
Short 'n Easy Cases



Reducing to Vertical U Perm



Reducing to Horizontal CCW U Perm (From Pi)



Reducing to Horizontal CW U Perm (From Pi)



Sz Sy2S (r)



Syz Sy2S (r)



Sy'z Sy2S (r)



Sy2z Sy2S (r)



Hz Sy2S (o)



Hyz Sy2S (o)



Hy'z Sy2S (o)



Hy2z Sy2S (o)

Peanut + Triple Sledge Case



Sy SS (b)



Sy' SS (o)



Hy' SS (g)



Hy SS (r)

Pi + Easy Horizontal U Perms



SySyS (b)



Sy'Sy'S (g)



Hy'Hy'H (b)



HyHyH (g)

Peanut + Horizontal U Perms



SySyH (g)



Sy'Sy'H (r)



SyHyH (r)



Sy'Hy'H (g)



Hy'Hy'S (o)



HyHyS (b)



Hy'Sy'S (b)



HySyS (o)

Pi + Z Perm Conjugates



Sy'SyH (g)



Sy'Sy'H (b)



SyHy'H (g)



Sy'HyH (b)

Peanut + Z Perm Conjugates



SySy'S (g)



Sy'SyS (r)



Hy'HyH (o)



HyHy'H (b)



SyHy'S (r)



Sy'HyS (g)



HySy'H (o)



Hy'SyH (b)

Peanut + U Perm Conjugates



SySy2H (o)



Sy'Sy2H (b)



SyHy2S (o)



Sy'Hy2S (b)



HyHy2S (g)



Hy'Hy2S (r)



HySy2H (r)



Hy'Sy2H (g)

Actual Pure Pi and U perm



SSy2z Sy2S (o)



SSz' Sy2S (o)



SSy2z Sy2S (o)



SSy2z' Sy2S (o)



SSy Sy2S (o)



SSy' Sy2S (o)

Reduce to Pure Pi (From Peanut)



Sy2Hy SS (r)



Sy2Hy' SS (g)



Hy2Sy' SS (b)



Hy2Sy SS (o)

Reduce to Triple Sledge (From Pi)



Sy SSS (r)



Sy' SSS (r)



Hy' SSS (o)



Hy SSS (o)

Peanut + Wat Perm



Sy2SySyS (b)



Sy2Sy'Sy'S (o)



Sy2Sy'HyH (r)



Sy2SyHyH (g)



Sy2SySyH (r)



Sy2Sy'Sy'H (g)



Sy2SyHyS (o)



Sy2Sy'Hy'S (b)

Peanut + Swirl Perm



SySy'SyS (r)



Sy'SySy'S (g)



Hy'HyHy'H (b)



HyHy'HyH (o)



SyHy'SyH (g)



Sy'HySy'H (r)



Hy'SyHy'S (o)



HySy'HyS (b)

Pi + Swirl Perm



SySy'SyH (o)



Sy'SySy'H (o)



Hy'HyHy'S (r)



HyHy'HyS (r)



SySy'HyS (g)



Sy'SyHy'S (b)



Hy'HySy'H (g)



HyHy'SyH (b)

Pi + Wat Perm



Sy2HySyS (g)



Sy2Hy'Sy'S (b)

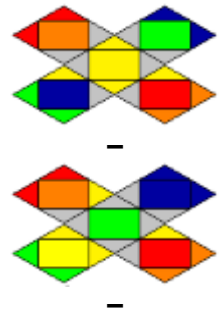
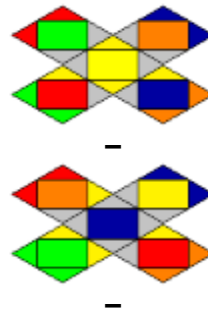
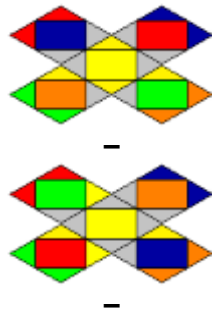
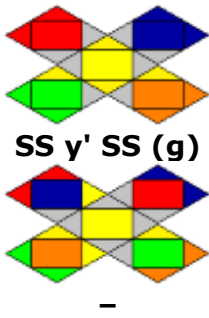


Hy2Sy'Hy'H (g)



Hy2SyHyH (b)

Suck Cases



L5C

