Square 1

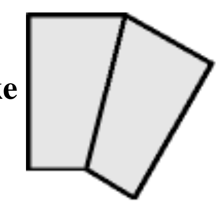
A Rubik's Cube on Acid

"*Ohhh... I'm sooooo wasted!*"

Transform the Puzzle into a Cube

Step I: Get the puzzle into 3 distinct layers

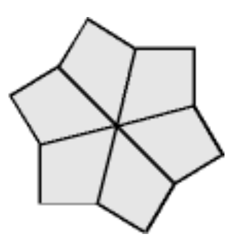
Square 1 is a mixture of small wedges, large wedges, and two **trapezoids**. The two **trapezoids** belong in the equator (middle layer) of the puzzle, and must be joined together:

like this:  **or like that:** 

Once the trapezoids are joined together, the puzzle is automatically at 3 distinct layers. This is way too easy, for two reasons:

- No matter how badly the puzzle is scrambled, you are (at the most) only ONE move away from getting there.
- At that point, it was the ONLY move you could have made here anyway!

Step II: Try to fill one layer with 6 large wedges



Once the puzzle is at 3 distinct layers, it can appear as one of many (90, to be exact) geometric shapes. There are just too many permutations to memorize, so the easiest way I can think of is to follow this simple rule:

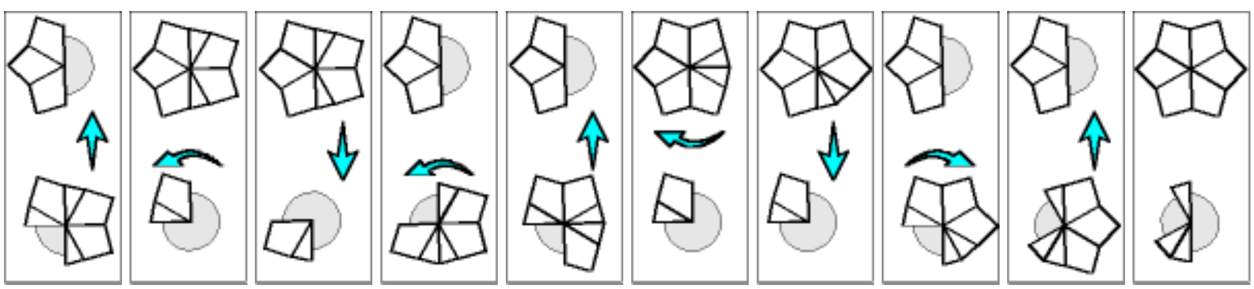
Fill an outside layer (top or bottom) with 6 large wedges. This will force the other side to have 2 large wedges and 8 small wedges. There are only 5 possible combinations for that mixture, therefore there will be only 5 tranformations to memorize afterwards.

This is not as hard as you think, but if you are still having trouble then I suggest you visit...

Christian Eggermont's Square 1 Page

...where you can start from any absurd shape to the final cube, one step at a time. The charts do not expect you to memorize the steps, but will help you get through that first obstacle. Besides, the geometric shapes are rather beautiful to look at.

As you become more familiar with and accustomed to the puzzle, then you will find it easier to get six large wedges on one layer all by yourself without the charts, and all you really had to do was play around with it for a few days. The **key** is to get three large wedges next to each other on both layers. Here is one **example:**

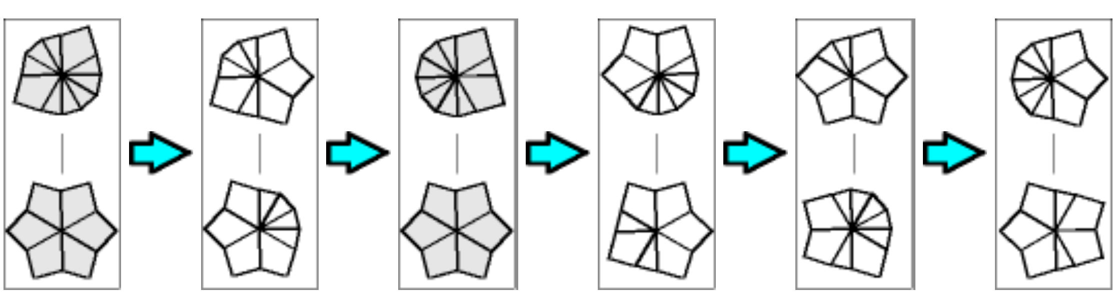


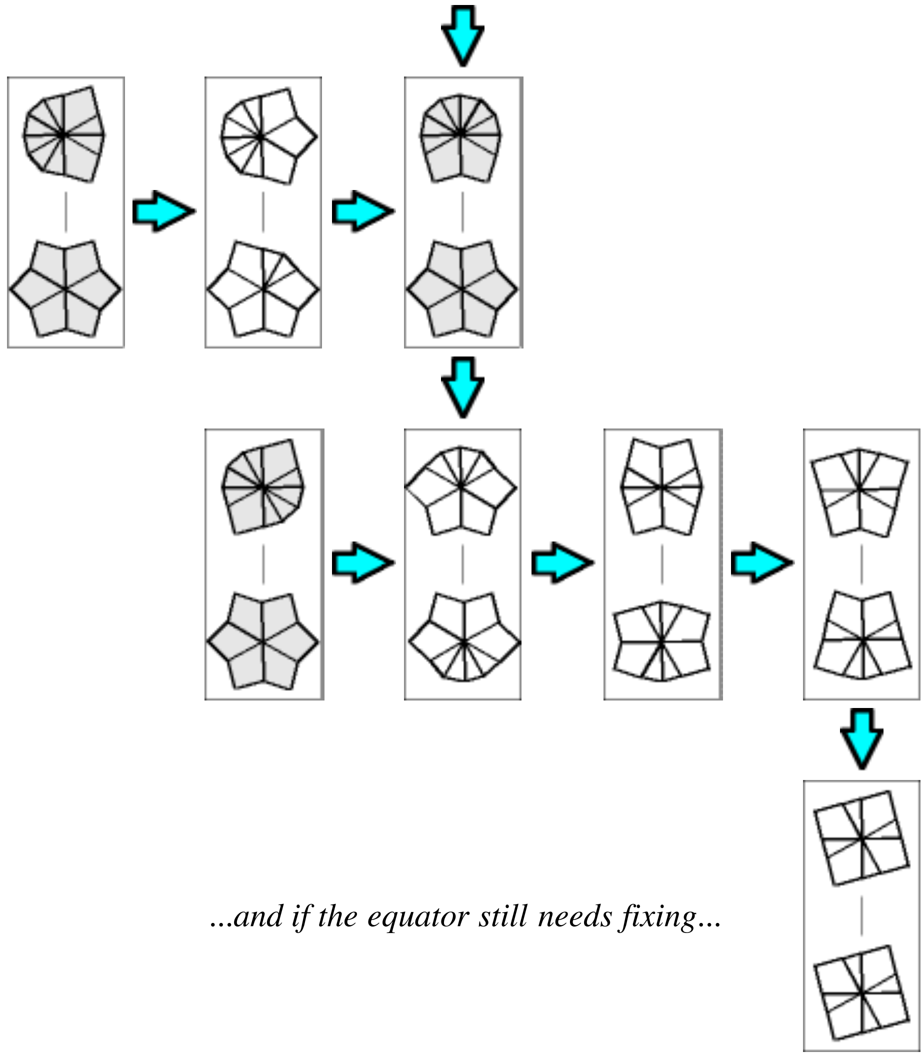
Note: There is no need to memorize the above table; it is merely an ***example****.*

Step III: Transform the puzzle into a cube

Once one layer is filled with 6 large wedges, then it is time to **memorize** the chart below to transform the puzzle to its final shape, the CUBE; or at least **practice** it enough so it becomes second nature.

The 5 possible **starting postions** are shaded in **gray**. Find your *pole position* and follow the arrows until you arrive at the *finish line*. To make a move, start off by setting the top and bottom layers just like one of the diagrams, then give the entire right side a twist. After that, turn the top and/or bottom layer so that they match the next diagram, before doing the twist again.





NOTE: the **gap** between the layers represents the **front edge**;
i.e., the front edge of the top layer points down,
the front edge of the bottom layer points up,
and finally, the "line-thingy" in the middle is the "slice" in the equator.

Notation

There are two different types of notation here. The first is what I call **Descriptive Notation**, which seems to be the accepted notation in most Square-1 web sites, so there is no sense for me to invent yet another one. The other was created by **Jaap Scherphuis**, which I prefer mainly because of its clarity and simplicity. Besides, it's a lot easier to type! Anyway, you only need to choose one of them.

Descriptive Notation:	***A SINGLE VOLUME OF MORAL I N S T R U C T I O N***	**Jaap Scherphuis Notation:**

t+	Rotate the **top layer 30 degrees** (1/12 of-a-turn) **clockwise.** *Note: a small wedge is 30 degrees wide.*	**(1,0)**
t+2	Rotate the **top layer 60 degrees** (1/6 of-a-turn) **clockwise.** *Note: a large wedge is 60 degrees wide.*	**(2,0)**
t+3	Rotate the **top layer 90 degrees** (1/4 of-a-turn) **clockwise.**	**(3,0)**
t6	Rotate the **top layer 180 degrees** (half-way around)	**(6,0)**
t-3	Rotate the **top layer 90 degrees** (1/4 of-a-turn) **counter-clockwise.**	**(-3,0)**
t-2	Rotate the **top layer 60 degrees** (1/6 of-a-turn) **counter-clockwise.**	**(-2,0)**
t-	Rotate the **top layer 30 degrees** (1/12 of-a-turn) **counter-clockwise.**	**(-1,0)**
R	Rotate the entire **RIGHT SIDE 180 degrees** (half-way around).	**/**
b+	Rotate the **bottom layer 30 degrees** (1/12 of-a-turn) **clockwise.**	**(0,1)**
b+2	Rotate the **bottom layer 60 degrees** (1/6 of-a-turn) **clockwise.**	**(0,2)**
b+3	Rotate the **bottom layer 90 degrees** (1/4 of-a-turn) **clockwise.**	**(0,3)**
b6	Rotate the **bottom layer 180 degrees** (half-way around)	**(0,6)**
b-3	Rotate the **bottom layer 90 degrees** (1/4 of-a-turn) **counter-clockwise.**	**(0,-3)**
b-2	Rotate the **bottom layer 60 degrees** (1/6 of-a-turn) **counter-clockwise.** *Note: a large wedge is 60 degrees wide.*	**(0,-2)**
b-	Rotate the **bottom layer 30 degrees** (1/12 of-a-turn) **counter-clockwise.** *Note: a small wedge is 30 degrees wide.*	**(0,-1)**

Basically you get the idea: **t** for top and **b** for bottom; **plus** for clockwise and **minus** for counter-clockwise, followed by the number of **increments** (1,2,3... etc.); and finally **R** for turning the right side. When using the **JS** notation, you can combine both top and bottom moves within a single set of parenthesis. To learn more about Jaap Scherphuis' notation, I strongly suggest that you visit his Square 1 page:

[Jaap's Square 1 Page]

NOTE: A **clockwise** move for bottom layer spins in the ***opposite*** direction of a clockwise move for the top layer. Imagine holding a jar with the lid on the top. To close the jar, you must turn the lid clockwise, twisting it to the **left**. Now imagine holding a jar upside-down. To close the same jar, you must also turn the lid clockwise, but now it twists to the **right** instead.

NOTE continued: The same goes for the **counter-clockwise** move; as they also spin in ***opposite*** directions for both top and bottom layers. To remove the lid, you must turn it counter-clockwise. If you are holding the jar right-side-up, then you must twist to the **right**. If you are holding the jar upside-down, then you must twist to the **left**.

I know this is all very confusing, but just pretend that for every clockwise move, you're screwing a lightbulb in; and for every counter-clockwise move, you're unscrewing a lightbulb out.

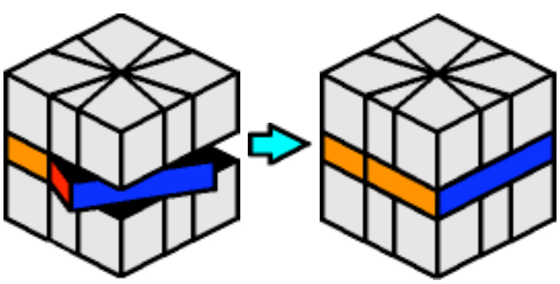
Because the **Square 1** puzzle has a tendency to mutate into so many bizzare shapes, I will try to keep things as **orthogonal** as possible; in other words, I *almost* promise to keep the top and bottom layers square throughout the solution. I also *almost* promise to keep each move itself at right-angle increments, with the minor exception of the +1 or -1 moves that lead and trail for each sequence.

The First Sequence

Once the puzzle is transformed into a cube, you only have to memorize ***ten sequences*** *to solve the colors, starting right NOW...*

Fix the equator:

Now that we are aquainted with the notation, it is time to learn your **first lesson** by solving the equator.



	Desc. notation:	JS notation:
	b- R t6 R t6 R b+	**(0,-1)/(6,0)/(6,0)/(0,1)**

This is probably the most important move to memorize, as there will be times when the equator accidentally gets out of shape. After the move is over, the equator is square again, and the top and bottom layers are left exactly as they were before.

A Note of Thanks

Before we continue, I must give credit where credit is due:

First of all, many kudos go to **Jaap Scherphuis** for helping me **polarize** the cube, which you will have to do later on yourself during the course of this solution. Just as he is indebted to Robert Richter for transforming the puzzle into cube, I feel equally indebted to him for teaching me how to get the the puzzle from **odd parity** to **even parity**. In the meantime, you should seriously consider visiting his home page; the single most comprehensive puzzle site on the web:

Jaap's Puzzle Page

Another round of applause goes to **Christian Eggermont**, the web-site that started it all! A decade ago, this was the **only** puzzle site available at the web. Today, it is still considered as the mecca for many cubists:

Christian Eggermont's Puzzle Page

NEXT: Solve the Top Corners

@ Transformation @ Top Corners @ Bottom Corners
@ Top Edges @ Polarization @ Bottom Edges

Return to Mathematica

-

Square 1

Solve the Top Corners

During this section, you must learn 3 different sequences.

By now, we know that the 2 trapezoids belong in the equator. But what about the wedges? If you haven't guessed by now, the answer is:

- **Large Wedge = Corner (top or bottom)**
- **Small Wedge = Edge (top or bottom)**
- **Trapezoid = Equator (only)**

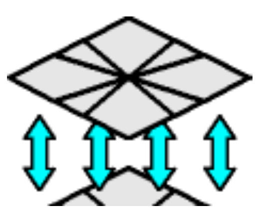
The **stategy** is to move up all the top corners to the top layer first; followed by placing them in their correct spots later.

Q: What color is the **TOP SIDE?**

A: WHITE; which means that the **bottom side** must be **green.** Square 1 seems to be committed to a single color scheme, so there's no sense going on (and on and on) about how to figure out what the colors are all by yourself. While we're at it, the **front** side is **orange**; and the **left** side is **yellow** (the same side that has the Square 1 logo). All the Square 1 web sites that I've seen unamonously agree that this is the universal color code.

Swap All 4 Corners from the Bottom to the Top

You always have the option to move up a white corner to the top layer one at a time; but if you have **3 or 4** white corners on the bottom layer, then consider using this shortcut. Afterwards/Otherwise, go to the next sequence.



Desc. notaion:

b- R t6 b6 R b+

JS notation:

(0,-1)/(6,6)/(0,1)

What the move really does: it swaps the entire top layer with the bottom layer; corners, edges and all. Right now we are concentrating on the corners, so we don't care about the fate of the edge pieces for now.

Move Up a Single Corner

When there are only **1 or 2** white corners on the bottom layer, then you have to use this sequence to move them up one at a time. Of course, if all 4 white corners are already on the top layer then you can skip this sequence altogether.

Desc. notaion:

t+ R b-3 R b-3 R b6 R t-

JS notation:

(1,0)/(0,-3)/(0,-3)/(0,6)/(-1,0)

Set Up:

- Top Layer: VACANT corner at front-right
- Bottom Layer: corner TO GO at front-right

Don't worry about getting the top corner in the correct spot for now, just climb it up! We'll worry about fixing them **later**.

Fix the Top Corners:

All 4 top corners are now on the top layer. The tops of all 4 corners are white, but what about their sides? At this point, there are 3 possibilities:

- No side has matching colors
- Only 1 side has matching colors
- All 4 sides have matching colors

Now that we know the status of the top corners, it is time to fix them. Of course, if all 4 sides have matching colors, then the top corners are already solved.

Case I: No Side has Matching Colors

SET UP:	Do the Move:	Result:
None 	Desc. notation: **t+ R b-3 R b+3 R b-3 R b-3 R b6 R t-** JS notation: **(1,0)/(0,-3)/ (0,3)/(0,-3)/ (0,-3)/(0,6)/(-1,0)**	 One side will have matching colors, so go to the next case.

Case II: Only 1 Side has Matching Colors		
SET UP:	**Do the Move:**	**Result:**
Rotate the top layer until the matching corners are both on the left side. 	Desc. notation: **t+ R b-3 R b+3 R b-3 R b-3 R b6 R t-** JS notation: **(1,0)/(0,-3)/ (0,3)/(0,-3)/ (0,-3)/(0,6)/(-1,0)**	 All 4 sides will have matching colors, so go to the next case.

Case III: All 4 sides have Matching Colors	
	The top corners are already solved!

	Now you can proceed to solve the **Bottom Corners**

What the Move Really Does:

In case you haven't noticed, the same move (for fixing the top corners) was used in both Case I and Case II; so just in case you are curious, here's what really happens:

Desc. notation:

t+ R b-3 R
b+3 R b-3 R
b-3 R b6 R t-

JS notation:

(1,0)/(0,-3)/
(0,3)/(0,-3)/
(0,-3)/(0,6)/(-1,0)

It swaps the adjacent right-side corners on the top layer, while swapping two opposite corners on the bottom layer

As you proceed to fix the top corners, the bottom corners become more and more scrambled; but you do not care about the fate of the bottom corners, since you are going to solve them next anyway.

NEXT: Solve the Bottom Corners

@ Transformation @ Top Corners @ Bottom Corners
@ Top Edges @ Polarization @ Bottom Edges

Return to Mathematica

-

Square 1

Solve the Bottom Corners

During this section, you only have to memorize one sequence.

Guess what? Because all 4 of the top corners are on the top layer by now, then all 4 bottom corners are already forced on the bottom layer, where they belong anyway. And as we all know by now, the bottom side is **green**.

Q: Who lives in a pineapple under the sea?
A: SPONGEBOB SQUAREPANTS!

Fix the Bottom Corners

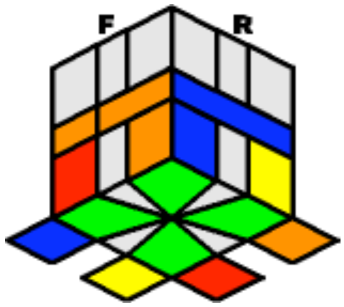
Because all 4 bottom corners are already forced on the bottom layer, all we have to do is shift them around to their correct places. The bottoms of all 4 corners are green, but what about their sides? At this point, there are 3 possibilities:

- No side has matching colors
- Only 1 side has matching colors
- All 4 sides have matching colors

Now that we know the status of the bottom corners, it is time to fix them. Of course, if all 4 sides have matching colors, then the bottom corners are already solved.

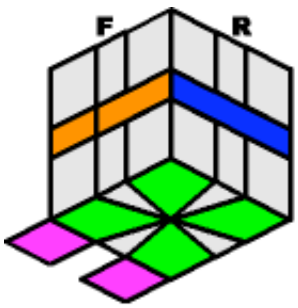
Case I: No Side has Matching Colors

SET UP:	Do the Move:	Result:
	Desc. notation:	

None  	**b- R t+3 R** **t-3 R t+3 R** **t+3 R t6 R b+** JS notation: **(0,-1)/(3,0)/** **(-3,0)/(3,0)/** **(3,0)/(6,0)/(0,1)**	 The bottom corners have one side with matched colors, but the top layer is temporarily scrambled. To fix both layers, we must do another step...
Now, rotate the bottom layer until the two matching sides are on the left:  	Desc. notation: **b- R t+3 R** **t-3 R t+3 R** **t+3 R t6 R b+** JS notation: **(0,-1)/(3,0)/** **(-3,0)/(3,0)/** **(3,0)/(6,0)/(0,1)**	 All 4 sides of the bottom layer match; the top and bottom corners are solved.

Case II: One Side has Matching Colors

SET UP:	**Do the Move:**	**Result:**
Rotate the bottom layer until the two matching sides are on the back:  	Desc. notation: **b- R t+3 R** **t-3 R t+3 R** **t+3 R t6 R b+**	

	JS notation: **(0,-1)/(3,0)/ (-3,0)/(3,0)/ (3,0)/(6,0)/(0,1)**	The bottom corners still have one side with matched colors, while the top layer becomes temporarily scrambled. To fix both layers, we must do another step...
Now, rotate the bottom layer until the two matching sides are on the left:  	Desc. notation: **b- R t+3 R t-3 R t+3 R t+3 R t6 R b+** JS notation: **(0,-1)/(3,0)/ (-3,0)/(3,0)/ (3,0)/(6,0)/(0,1)**	 All 4 sides of the bottom layer match; the top and bottom corners are solved.

Case III: All 4 sides have Matching Colors

The bottom corners are already solved!
Now you can proceed to solve the
Top Edges

What the Move Really Does:

In case you haven't noticed, the same sequence was used throughout this entire section; so just in case you are curious, here's what really happens:

Desc. notation:

b- R t+3 R

t-3 R t+3 R
t+3 R t6 R b+

JS notation:
(0,-1)/(3,0)/
(-3,0)/(3,0)/
(3,0)/(6,0)/(0,1)

It swaps two opposite corners on the top layer, while swapping the two adjacent right-side corners on the bottom layer.

You must always perform this sequence twice. As you solve the bottom corners, the top corners become scrambled after the first time, and then unscrambled after the second time. After the smoke clears, the bottom corners are finshed as well.

NEXT: **Solve the Top Edges**

@ Transformation @ Top Corners @ Bottom Corners
@ Top Edges @ Polarization @ Bottom Edges

Return to Mathematica

-

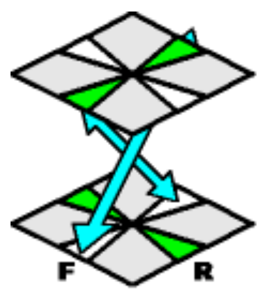
Square 1

Solve the Top Edges

During this section, you must learn 3 other sequences.

Solve Two of the Top Edges:

It is time to introduce yourself to a *new* move. The move will be used throughout the first section of this page:

Desc. Notation:	**b- R b+** **t+ R t-**		**What the move does:** It swaps the front and back pair of edges between the top and bottom layers; not directly above-and-below, but in a "criss-cross" way.
JS Notation:	**(0,-1)/(1,1)/(-1,0)**		

Your job is to get 2 adjacent (side-by-side) edges, one at a time, to their **correct spots** on the top layer; this is very important. There are four color pairs to choose from (of course, each top edge piece also has a white side added to it):

- yellow / orange
- orange / blue
- blue / red
- red / yellow

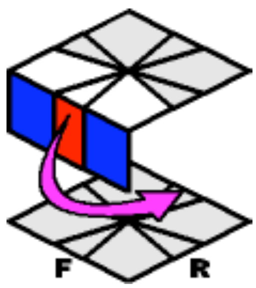
~~~~ Step I ~~~~

Start off by solving the **first** top edge. You can choose any color you like...

MOVE UP:	Set up:	Do the move:	Result:
		Desc. notation:	

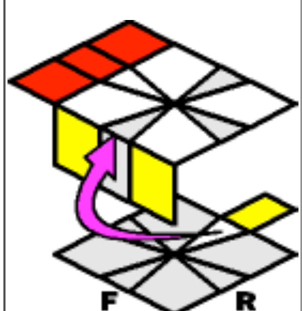
 	Top layer: vacant edge on front. Bottom layer: edge to-go in back.	**b- R b+** **t+ R t-** JS notation: **(0,-1)/(1,1)/(-1,0)**	Now at least one top edge is solved. Go ahead and solve the **next** top edge (see Step II).

The **first** top edge you chose may already be in the top layer but on the wrong side. You must knock it down before moving it back up to the correct spot...

KNOCK DOWN:	Set up:	Do the move:	Result:
 	Top layer: edge to-go on front. Bottom layer: don't care.	Desc. notation: **b- R b+** **t+ R t-** JS notation: **(0,-1)/(1,1)/(-1,0)**	The top edge is now on the bottom layer. Go **back** and move it up to the top layer.

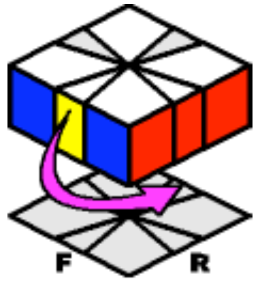
~~~~ Step II ~~~~

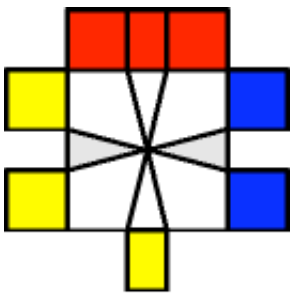
Now you can solve the **next** top edge. With exception, you can choose any color you like, as long as that color is **adjacent** to the first solved edge...

MOVE UP:	Set up:	Do the move:	Result:
 	Top layer: vacant edge on front. Bottom layer: edge to-go in back.	Desc. notation: **b- R b+** **t+ R t-** JS notation: **(0,-1)/(1,1)/(-1,0)**	**Bingo!** Now there are two adjacent edges solved on the top layer.

The **next** top edge you chose may already be in the top layer but on the wrong side. You must knock it down before moving it back up to the correct spot...

KNOCK DOWN:	Set up:	Do the move:	Result:
		Desc. notation:	

 F R	Top layer: edge to-go on front. Bottom layer: don't care.	**b- R b+** **t+ R t-** JS notation: **(0,-1)/(1,1)/(-1,0)**	The top edge is now on the bottom layer. Go **back** and move it up to the top layer.



ROAD BLOCK! You can choose any color for the **first** top edge, but you are limited to select the color of the **next** top edge. For example, look at the diagram... The **red** top edge is already in place. The **yellow** top edge is also on the top layer but on the wrong side; which means it needs to be knocked down and moved back up later. However, if you knock down the **yellow** edge, the **red** edge goes down as well. Because the two edges are on the same layer and **opposite** of each other, you cannot pair them together side-by-side using this method. Luckily, you still have another top edge to choose from, the **blue** one.

Place the other Two Top Edges:

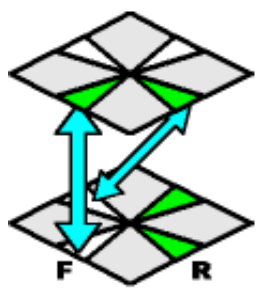
Now it is time to introduce yourself to *another* new move. The move will be used throughout the second section of this page:

Desc. Notation:

b- R b-3 R b+
t+ R t6 R t-
b- R b+3 R b+
t+ R t6 R t-

JS Notation:

(0,-1)/(0,-3)/(1,1)/
(6,0)/(-1,-1)/(0,3)/
(1,1)/(6,0)/(-1,0)

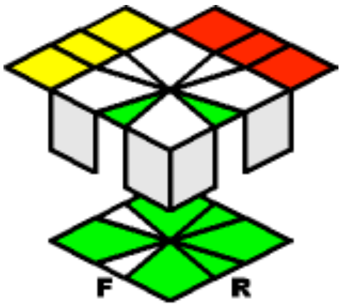
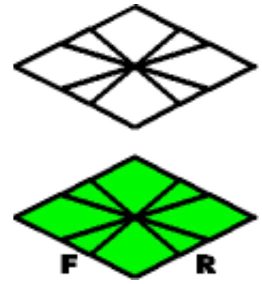


What the move does:
It swaps the top-front edge with the bottom-front edge, and swaps the top-right edge with the bottom-left edge.

This time, you do **not** have to worry about moving the other top edges to their correct spots; just get them both up in the top layer! However, it is important that you do **not** disturb the two original top edges that are already finished.

~~~~ Case I ~~~~

SET UP:	Do the move:	Result:

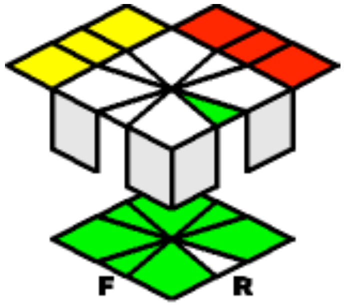
Top Layer: original edges to the left and back. Bottom Layer: white edges to the left and front. 	Desc. notation: **b- R b-3 R b+** **t+ R t6 R t-** **b- R b+3 R b+** **t+ R t6 R t-** JS notation: **(0,-1)/(0,-3)/(1,1)/(6,0)/(-1,-1)/(0,3)/(1,1)/(6,0)/(-1,0)**	 **Bingo!** All 4 top edges are on the top layer!

~~~~ Case II ~~~~

SET UP:	Do the move:	Result:
Top Layer: original edges to the left and back. Bottom Layer: white edges to the left and right. 	Desc. notation: **b- R b-3 R b+** **t+ R t6 R t-** **b- R b+3 R b+** **t+ R t6 R t-** JS notation: **(0,-1)/(0,-3)/(1,1)/(6,0)/(-1,-1)/(0,3)/(1,1)/(6,0)/(-1,0)**	 Go to **Case IV.**

~~~~ Case III ~~~~

SET UP:	Do the move:	Result:
Top Layer: original edges to the left and back; with the "extra" white edge at the front. Bottom Layer: single white edge at the right.	Desc. notation: **b- R b-3 R b+** **t+ R t6 R t-** **b- R b+3 R b+**	

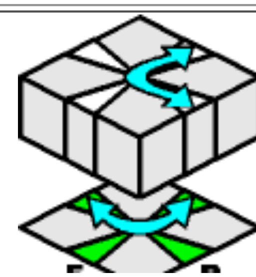
	**t+ R t6 R t-** JS notation: **(0,-1)/(0,-3)/(1,1)/ (6,0)/(-1,-1)/(0,3)/ (1,1)/(6,0)/(-1,0)**	 Go to **Case I.**

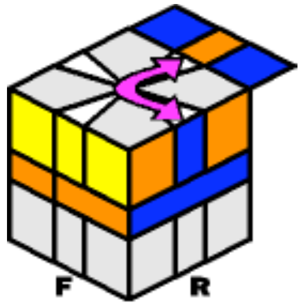
~~~~ Case IV ~~~~

SET UP:	Do the move:	Result:
Top Layer: original edges to the left and back; with the "extra" white edge at the right. Bottom Layer: single white edge at the back. 	Desc. notation: **b- R b-3 R b+ t+ R t6 R t- b- R b+3 R b+ t+ R t6 R t-** JS notation: **(0,-1)/(0,-3)/(1,1)/ (6,0)/(-1,-1)/(0,3)/ (1,1)/(6,0)/(-1,0)**	 Go to **Case I.**

Fix the Top Edges:

By now, all 4 top edges are on the top layer, with two adjacent edges already in place. The other two may also be in place, but if they are not then swap them by using *yet another* new move:

SET UP:	Do the move:	Result:	What the move does:
Top Layer: edges to be swapped at the right and the back. Bottom Layer: don't care.	Desc. notation: **b- R b-3 R b+ t+ R b+3 R t- b- R b+ t+ R t- JS**		

	**notation:** **(0,-1)/(0,-3)/ (1,1)/(0,3)/ (-1,-1)/(1,1)/(-1,0)**	The top layer is **solved.**	 It swaps the back and right edges on the top layer, while swapping the back and left edges on the bottom layer.

What's Next?

Normally, you would expect to go ahead and solve the **Bottom Edges.** Unfortunately, Square-1 is the sort of puzzle that can bring up many strange and eerie surprizes. The best thing for you to do is visit the **polarizing** page next, and then check to find out if you need to put yourself through this kind of carnage. Afterwards/Otherwise, you can go on to solve the **bottom edges.**

So check and see whether or not you have to...

Polarize the Cube

@ Transformation @ Top Corners @ Bottom Corners
@ Top Edges @ Polarization @ Bottom Edges

Return to Mathematica

-

Square 1

Polarize the Puzzle

During this section, you must memorize 2 more moves; the POLARIZING sequence and the FIXING sequence.

First of all, many thanks to **Jaap Scherphuis** for helping me out through these tough times of **polarization, parity,** and (what HE calls) **permutations** concepts. Just as he is indebted to Robert Richter for transforming the puzzle into cube, I feel equally indebted to him for teaching me how to easily permutate the puzzle from **odd to even.** Who knows? Maybe Square-1 is the only puzzle that was solved via the world-wide-web. To learn more about Jaap Scherphuis' theories, I strongly suggest that you visit his Square-1 page:

Jaap's Square 1 Page

On the Rubik's Cube, it is impossible for a single pair of edges to become swapped . However, on Square 1, it is *indeed* possible for this to happen. If only *a single pair* of edges need swapped, then puzzle is at **ODD** parity; ...**EVEN** parity is when *two* edge pairs need swapped. The move currently used to solve the bottom edges can only handle cubes at even parity, so the answer is to convert the parity from ODD to EVEN. I call it **polarizing,** but you can call it anything you want.

Does the puzzle need to be polarized?

You have to polarize your puzzle **if and only if** the parity is **odd.** The next question is:

How can you tell what the parity is?

Assuming that the top layer and bottom corners are finished, the clue lies in the unfinished bottom edges. Look at the bottom side; there are 10 total possibilities...

EVEN PARITY / bottom view

Parity = 2

Parity = 2

Parity = 2

Parity = 2

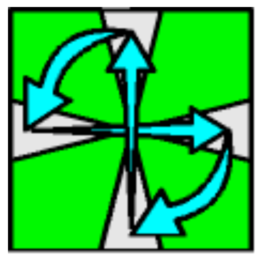
As you can see, parity has nothing to do with how many *edges* are scrambled, but instead how many *edge pairs* are swapped. If the bottom edges look like any of the 4 examples above, then the parity is already **even;** so do **not** by any means polarize your cube... instead, **bypass** this page and go directly to the next step:

Solve the Bottom Edges

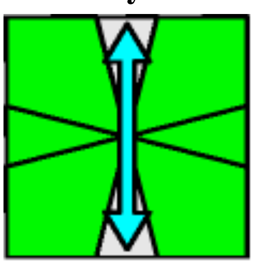
ODD PARITY / bottom view

Parity = 1

Parity = 3

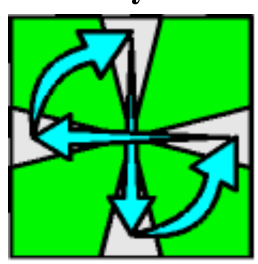


Parity = 3



Parity = 1

Parity = 3



Parity = 3

Once again, parity has nothing to do with how many *edges* are scrambled, but instead how many *edge pairs* are swapped. If the bottom edges look like any of the 6 examples above, then the parity is **odd;** which means you **absolutely** have to **polarize** your cube... so hold your breath and get ready for battle.

The Polarizing sequence:

SET UP: Make sure that the *front sides* of the top and bottom corners are *orange*.

Desc. Notation: JS Notation:

R t-3 b-3 R t-2 b- R	**/(-3,-3)/(-2,-1)/**
t-2 b-2 R b-2 R t-2 b-2	**(-2,-2)/(0,-2)/(-2,-2)**
R t-2 b- R t+3 b+3 R	**/(-2,-1)/(3,3)/**

*Click **here** to see the moves in graphical notation.*

While you do this move, you will notice that the puzzle will morph into the bizzarre shape it was at when you took it out of the box, causing an intense experience of deja-vu. After the move, the puzzle is restored to a cube again.

~~~~The Result~~~~

Oh No!

Can it ever be fixed?!? As it turns out, the fix is a rather simple sequence... but before doing the next move, check and make sure that the front sides of the *top* layer corners are *orange*, while the front sides of the *bottom* layer corners are *yellow*. (Ignore the colors of the edge pieces).

Fixing Sequence:

Desc. Notation:	JS Notation:
b- R t-3 b-3 R b+	**(0,-1)/(-3,-3)/**
t+ R t-3 b-3 R t+2	**(1,1)/(-3,-3)/(2,0)**

*Click **here** to see the moves in graphical notation.*

Now the top layer and bottom corners are solved again, and the cube is polarized at **even parity** to boot! The only pieces that are still remain scrambled are the bottom edges.

Now What?

If you did the polarizing and fixing moves correctly, the top layer and bottom corners will be solved again. Only the bottom edges are still scrambled, and because the cube is now at **even partity,** you can go ahead and...

Solve the Bottom Edges

Note: if you screwed up somehow, you'll have to start all over again; but chances are that the cube became correctly polarized somewhere along the way, so you can probably skip this section the next time around.

@ Transformation @ Top Corners @ Bottom Corners
@ Top Edges @ Polarization @ Bottom Edges

Return to Mathematica

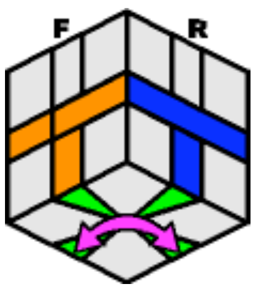
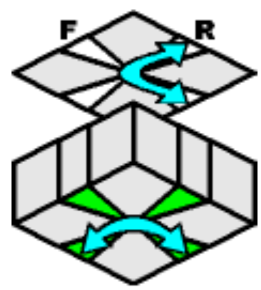
-

Square 1

Solve the Bottom Edges

During this section, you only have to use one sequence of moves.

Only one sequence is used throughout this page, one that has already been used before; so here is the RERUN...

<table>
<tr><th>What we want the move to do:</th><th>THE MOVE:</th><th>What the move really does:</th></tr>
<tr><td>To swap a single pair of edges on the bottom layer
</td><td>Desc. notation:
b- R b-3 R b+ t+ R b+3 R t- b- R b+ t+ R t-
JS notation:
(0,-1)/(0,-3)/ (1,1)/(0,3)/ (-1,-1)/(1,1)/(-1,0)</td><td>It swaps the back and left edges on the bottom layer, while swapping the back and right edges on the top layer.</td></tr>
</table>

As you can see, every time a pair of bottom edges are swapped, a pair of top edges also become swapped. Therefore, the sequence must be used again to re-fix the sacrificial top edges. The result? The edges on the bottom layer are swapped TWICE, which is why we had to **polarize** the cube to **even parity** before going ahead with this step.

You might be able to figure out how to individually swap the bottom edges until they are fixed; but just in case, a step-by-step outline was added.

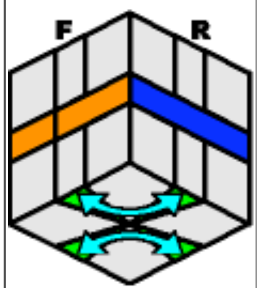
~~~~ Case I ~~~~
Clockwise Exchange

	**SET UP:** Rotate the bottom layer so that the 3 edges to be exchanged are on the back, left and right. In other words, get the "other edge" at the front.			
	Do the move:	**Rotate the bottom layer 1/4 of a turn clockwise**	**Do the move again:**	**Result:**
Desc. Notation:	**b- R b-3 R b+ t+ R b+3 R t- b- R b+ t+ R t-**	**b+3**	**b- R b-3 R b+ t+ R b+3 R t- b- R b+ t+ R t-**	The puzzle is **solved!**
JS Notation:	**(0,-1)/(0,-3)/ (1,1)/(0,3)/ (-1,-1)/(1,1)/(-1,0)**	**(0,3)**	**(0,-1)/(0,-3)/ (1,1)/(0,3)/ (-1,-1)/(1,1)/(-1,0)**	

~~~~ Case II ~~~~
Counter-Clockwise Exchange

	**SET UP:** Rotate the bottom layer so that the 3 edges to be exchanged are on the back, left and front. In other words, get the "other edge" at the right.			
	Do the move:	**Rotate the bottom layer 1/4 of a turn counter-clockwise**	**Do the move again:**	**Result:**
Desc. Notation:	**b- R b-3 R b+ t+ R b+3 R t- b- R b+ t+ R t-**	**b-3**	**b- R b-3 R b+ t+ R b+3 R t- b- R b+ t+ R t-**	The puzzle is **solved!**
JS Notation:	**(0,-1)/(0,-3)/ (1,1)/(0,3)/ (-1,-1)/(1,1)/(-1,0)**	**(0,-3)**	**(0,-1)/(0,-3)/ (1,1)/(0,3)/ (-1,-1)/(1,1)/(-1,0)**	

~~~~ Case III ~~~~
Adjacent Edge-Pair Swap

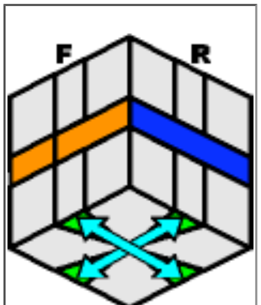


SET UP:

Rotate the bottom layer so that two of the edges that need to be swapped are at the back and left, while the other two are at the front and right.

	Do the move:	Rotate the bottom layer 1/2 of a turn around	Do the move again:	Result:
Desc. Notation:	**b- R b-3 R b+ t+ R b+3 R t- b- R b+ t+ R t-**	**b6**	**b- R b-3 R b+ t+ R b+3 R t- b- R b+ t+ R t-**	The puzzle is **solved!**
JS Notation:	**(0,-1)/(0,-3)/ (1,1)/(0,3)/ (-1,-1)/(1,1)/(-1,0)**	**(0,6)**	**(0,-1)/(0,-3)/ (1,1)/(0,3)/ (-1,-1)/(1,1)/(-1,0)**	

~~~~ Case IV ~~~~
Opposite Edge-Pair Swap



SET UP:

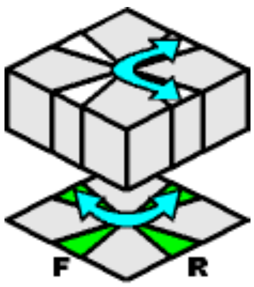
None (don't care).

	Do the move:	Rotate the bottom layer 1/4 of a turn at any direction	Do the move again:	Result:
Desc. Notation:	**b- R b-3 R b+ t+ R b+3 R t- b- R b+ t+ R t-**	**b+3 ~or~ b-3**	**b- R b-3 R b+ t+ R b+3 R t- b- R b+ t+ R t-**	The bottom layer has 3 edges that are still scrambled. If they need to be exchanged **clockwise,** then go to **Case I.** If they need to be
	(0,-1)/(0,-3)/ (1,1)/(0,3)/	**(0,3)**	**(0,-1)/(0,-3)/ (1,1)/(0,3)/**	

JS Notation:	**(1,1)/(0,3)/ (-1,-1)/(1,1)/(-1,0)**	**~or~ (0,-3)**	**(1,1)/(0,3)/ (-1,-1)/(1,1)/(-1,0)**	exchanged **counter-clockwise**, then go to **Case II.**

A Shortcut for the Experts:

This sequence does the exact same thing that other sequence did, but with fewer moves. The moves are are **not orthogonal,** but the top and bottom layer will always remain **square:**

Swap the back/right edges on the top layer, while swapping the back/left edges on the bottom layer:	Desc. Notation:	JS Notation:
	**b- R b-3 R b+ t+ R b+2 t- R b+**	**(0,-1)/(0,-3)/(1,1) /(-1,2)/(0,1)**

And now that the nightmare is finally over, you can live happily ever after.

THE END

Quick Summary of All Move Sequences:

Desc. Notation	JS Notation	Comments
b- R t6 R t6 R b+	**(0,-1)/(6,0)/(6,0)/(0,1)**	Fixes the Equator
b- R t6 b6 R b+	**(0,-1)/(6,6)/(0,1)**	Swaps the top and bottom layers
t+ R b-3 R b-3 R b6 R t-	**(1,0)/(0,-3)/(0,-3)/(0,6)/(-1,0)**	Moves up a single corner to the top
t+ R b-3 R b+3 R b-3 R b-3 R b6 R t-	**(1,0)/(0,-3)/ (0,3)/(0,-3)/ (0,-3)/(0,6)/(-1,0)**	Fixes the top corners
b- R t+3 R	**(0,-1)/(3,0)/**	

t-3 R t+3 R t+3 R t6 R b+	**(-3,0)/(3,0)/ (3,0)/(6,0)/(0,1)**	Fixes the bottom corners
b- R b+ t+ R t-	**(0,-1)/(1,1)/(-1,0)**	Places the first two top edges
b- R b-3 R b+ t+ R t6 R t- b- R b+3 R b+ t+ R t6 R t-	**(0,-1)/(0,-3)/(1,1)/ (6,0)/(-1,-1)/(0,3)/ (1,1)/(6,0)/(-1,0)**	Places the last two top edges
b- R b-3 R b+ t+ R b+3 R t- b- R b+ t+ R t-	**(0,-1)/(0,-3)/ (1,1)/(0,3)/ (-1,-1)/(1,1)/(-1,0)**	Fixes the top and bottom edges
R t-3 b-3 R t-2 b- R t-2 b-2 R b-2 R t-2 b-2 R t-2 b- R t+3 b+3 R	**/(-3,-3)/(-2,-1)/ (-2,-2)/(0,-2)/(-2,-2) /(-2,-1)/(3,3)/**	Polarizing move
b- R t-3 b-3 R b+ t+ R t-3 b-3 R t+2	**(0,-1)/(-3,-3)/ (1,1)/(-3,-3)/(2,0)**	Polarizing fix

---

@ Transformation @ Top Corners @ Bottom Corners
@ Top Edges @ Polarization @ Bottom Edges

Return to Mathematica

-