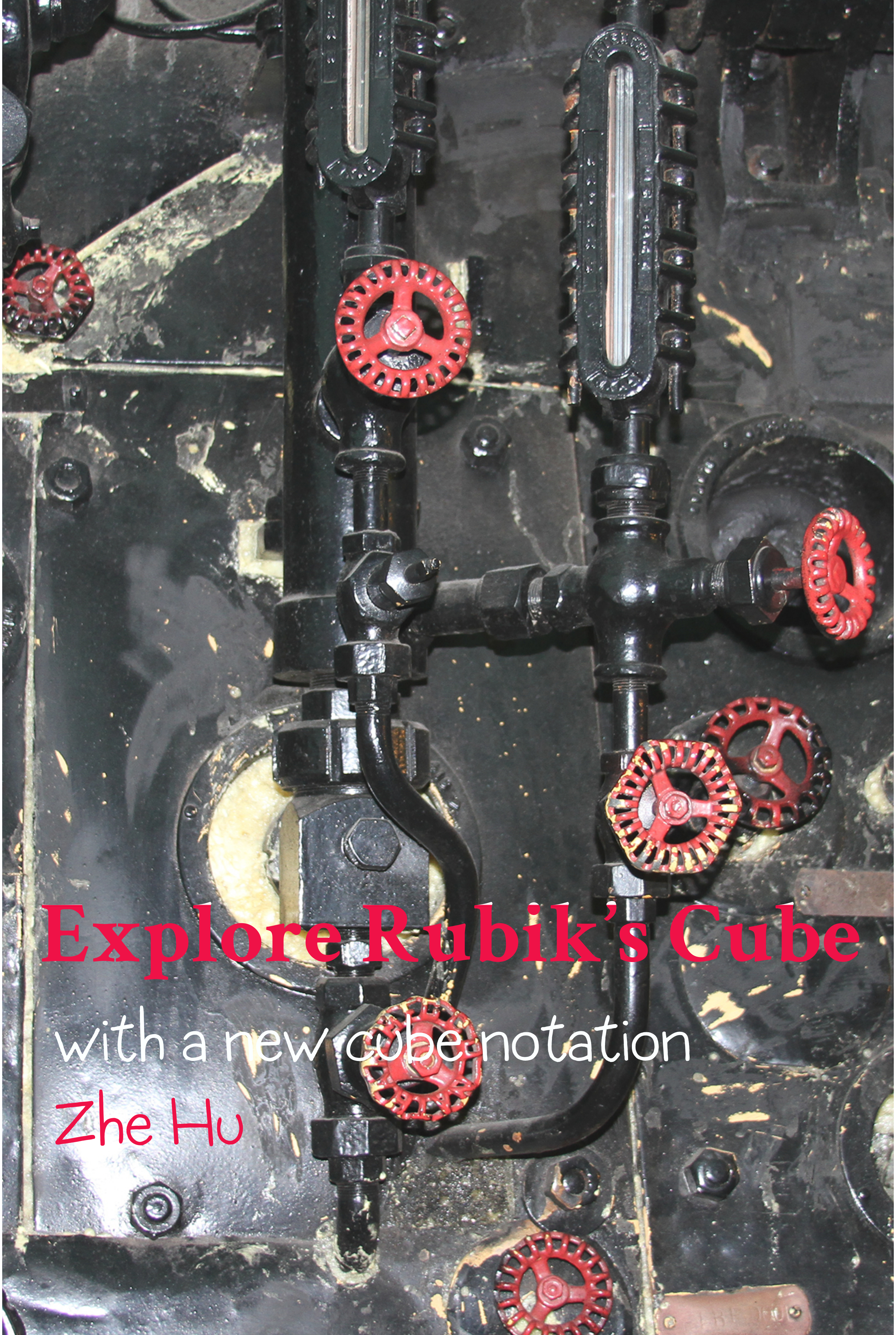


Explore Rubik's Cube

with a new cube notation

Zhe Hu

Explore Rubik's Cube

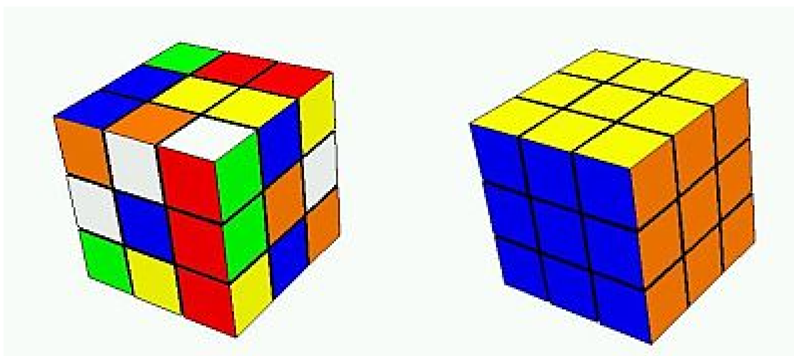
with a new cube notation

Zhe Hu

Contents

Preface

You have a Rubik's Cube. It is scrambled. You have an urge to solve the cube, that is to make each face a solid color.



Fortunately a small booklet comes with your Rubik's Cube purchase. (That booklet can also be downloaded from the official website.)

It introduces a 7-stage beginner's method to solving the cube. At each stage, you perform what's called an algorithm, consisting of a sequence of turning the cube faces. The algorithms are spelled out with sometimes long string of letters, each letter representing a clockwise or counter-clockwise turn of one of the six cube faces.

In addition to the booklet, a lot of helpful cube solving tutorials can be found through *youtube* or *google*.

However all these cube turn notations, algorithms, method acronyms, can be overwhelming at first. You may already have a sinking feeling that unless you spend a lot of hours learning and practicing, there is no chance you can learn to solve the cube by yourself.

Because even if you have managed to blindly follow some instructions or tutorials carefully, double-check every clockwise and counter-clockwise turn while executing an algorithm, and have gotten to the end with a solved cube. Congratulations! Now you wonder how you could possibly memorize these **long** algorithms in order to perform a live demo in front of your friends, for example.

So you either have to practice the algorithms regularly to make them muscle memory or you abandon this puzzle after a while.

This book gives you a third option. It teaches you how to solve the cube "intuitively", namely, by understanding cube algorithms through logic reasoning. We introduce a new cube notation that is easy to learn and it helps you visualize **what's going on**.

We will explore the layer-by-layer cube solving method, also called beginner's method, with the new cube notation.

The beginner's method won't prepare you for the sub-20 seconds speed cubing championship, it's possible that after working through this book, you find yourself a new passion for Rubik's Cube and start training for speed cubing. BTW, computer programs equipped with robotic arms have beaten humans long time ago, in term of how fast it can solve the cube.

We want to bring back the fun and amazement of playing Rubik's cube, not as a memorization tool, but as a tool to evoke deep logic thinking.

We hope that years after reading this book, if you get hold of a Rubik's Cube, you still know how to solve it, just from a few guiding principles. More importantly, once you learn the new cube notation, you can sketch and think about cube algorithms on paper, and even design your own algorithms. You can also explain and teach cube solving to others.

Imagine the following conversation:

> A: Bill, it's so cool. How do you do it?
>
> B: Well, this is the ZZ method. To complete F2L, you apply this algorithm R U2 R' . . .
>
> A: Okay, whatever. You are smart.

Or this conversation:

A: Bill, it's so cool. How do you do it?

B: Actually, Alice, it's not that difficult. Let's get a piece of paper. First you need to know a few principles …

A: (brightening) Great, let me have a try …

… …

A: Bill, you are wonderful. How do you figure all these out?

B: Well, many years ago, I read a book, called "Explore Rubik's Cube" …

It goes without saying, we prefer the second conversation.
Now let's get started.

Chapter 1

Getting started

> Keep It Simple, but not Simpler
>
> — KISS principle

1.1 Cube 101

There are 8 corner pieces, 12 edge pieces, and 6 center pieces on a cube. Out of the 6 faces, only 3 faces are visible in front of you at one time, while the other three faces are hidden from view. We designate these three visible faces as *front*, *up* (or *top*), and *right* faces respectively.

The color of the *up* face, for example, depends on how you hold the cube. Usually a standard Rubik's Cube pairs up colors of the opposite faces as: red–orange, white–yellow, green–blue.

In this book, we will mainly operate on the three visible faces: *front*, *right*, *up*.

By “operate”, we mean turning these faces 90-degree either clockwise or counter-clockwise. To turn a face, imagine yourself a plumber, clockwise turn is like closing a valve, counter-clockwise turn is like opening it (cover image of this book shows a lot of valves).

To solve a scrambled cube, one needs to turn the cube faces in certain order so that the same color pieces (edge, corner, and center pieces) are assembled on the same face in the end. It’s not easy to do.

Since the invention of Rubik’s Cube, cube solvers all over the world have discovered and refined many useful sequences of turning the cube, which are called algorithms. An algorithm is just a sequence of clockwise or counter-clockwise turns of certain cube faces. It is traditionally described as a string of letters, called “cube notation”, such as $R^{-1}U^{-1}RURU^{-1}R$.

Depending on the solving method, you may only need to execute a few algorithms to go from a scrambled cube to a solve one. The key is to know which algorithm to apply and how to execute the algorithm.

In general, the less algorithms to execute, the faster one can solve the cube. The problem is that these algorithms become very specific, that is each of them only handles certain cube pattern, but not others. So in order to solve the cube in all situations, a trained cube solver (human, not computer program) needs to memorize a large reservoir of cube algorithms to be able to quickly deal with different cube patterns.

That’s why there are a lot of different cube solving methods, all having different properties along these two dimensions:

1. the number of algorithms to learn
2. how fast one can solve the cube with these algorithms

Beginner's method, also called layer-by-layer method is one of the cube solving methods. It doesn't have a lot of algorithms to learn and one can solve the cube reasonably using this method.

More importantly the confidence, intuition and knowledge you build up learning this method can apply well to learning and practicing any other methods.

1.2 New cube notation

To help you understand and carry out the algorithms for this layer-by-layer method, we introduce a new graphic cube notation in this book.

Contrary to the traditional notation, we don't explicitly spell out the cube turns in letters, rather we choose to sketch out the result of each turn, that is the current state of the cube.

The idea is that, the trajectory (or orbit) of a single edge or corner cube piece is traced out by executing an algorithm in a meaningful way. By graphing that trajectory step-by-step in a sequence of sketches, we help you visualize **what's going on** with an algorithm.

Each sketch is usually a snapshot of the 3 cube faces (up, front, and right) with the essential cube pieces marked in numbers.

According to "cube theory" (i.e. group theory applied to the cube), each cube move can affect a number of pieces at the same time. By only focusing on one piece, we are missing a lot of actions. However we sometimes ignore these "side effects" on purpose, not because they are not important, but it helps us break a complex problem into manageable smaller pieces to solve.

Figure 1.1 shows an example of our cube sketch notation.

Let's explain how to read this new cube notation next.

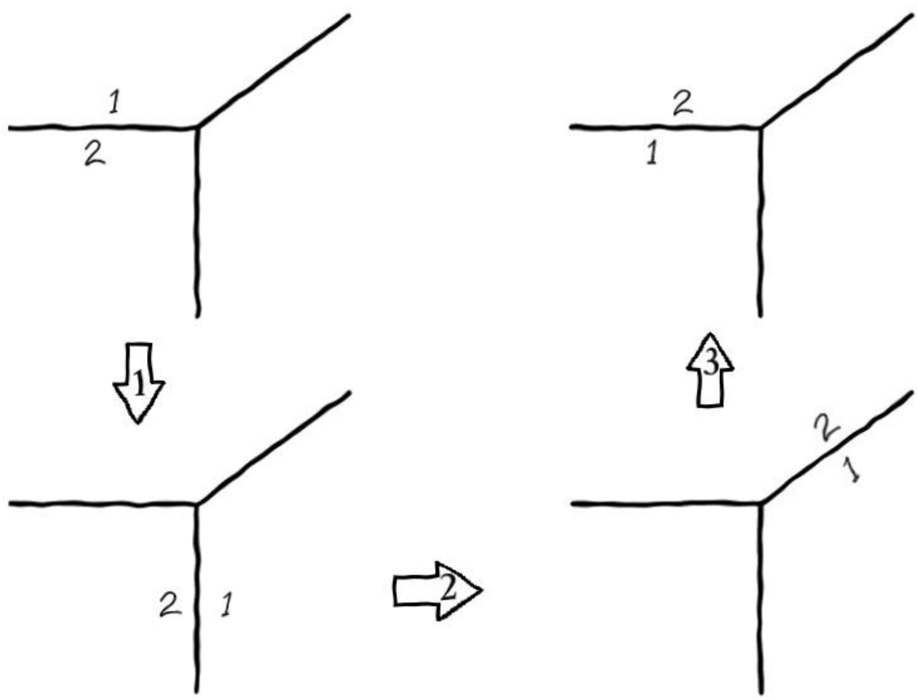


Figure 1.1: Simple edge flip algorithm.

1.3 Start with a few facts

Here are two intuitive facts about the cube.

> Fact 1: The center color of each face will never change by turning that face.

In fact the relative positions of six center pieces stay fixed. They determine the positions of the other edge and corner pieces of a solved cube.

In the meantime, the corner and edge pieces can move around by turning any one of the six faces. First, let's find a way to graph the movement of these edge pieces?

> Fact 2: Each edge piece has two color squares (marked as 1–2 in Figure 1.2). On each 90-degree turn, the color square on the turning face will **stay on that face**, whereas the other square will **jump an edge**.

With these two facts in mind, let's read our new cube notation in Figure 1.2.

First of all, since the center pieces don't change, there is no need to mark them in the sketch. A lot of edge and corner pieces do move around after each cube turn, but we will stay focus on one piece at a time. The steps 1, 2, 3 with arrows corresponds to the numbered descriptions below:

1. A clockwise turn of the front face. Piece **2** remains on the front face and piece **1** jumps an edge—from up face to right face.
2. A clockwise turn of the right face. This time **1** stays on the right face, because that's the turning face, and **2** jumps an edge—from front face to up face.
3. Another clockwise turn of the right face. Again **1** stays on the right face, and **2** jumps another edge. **2** falls off the cliff to the invisible back face.

After some practice, you should be able to "read" the sketches without the descriptions.

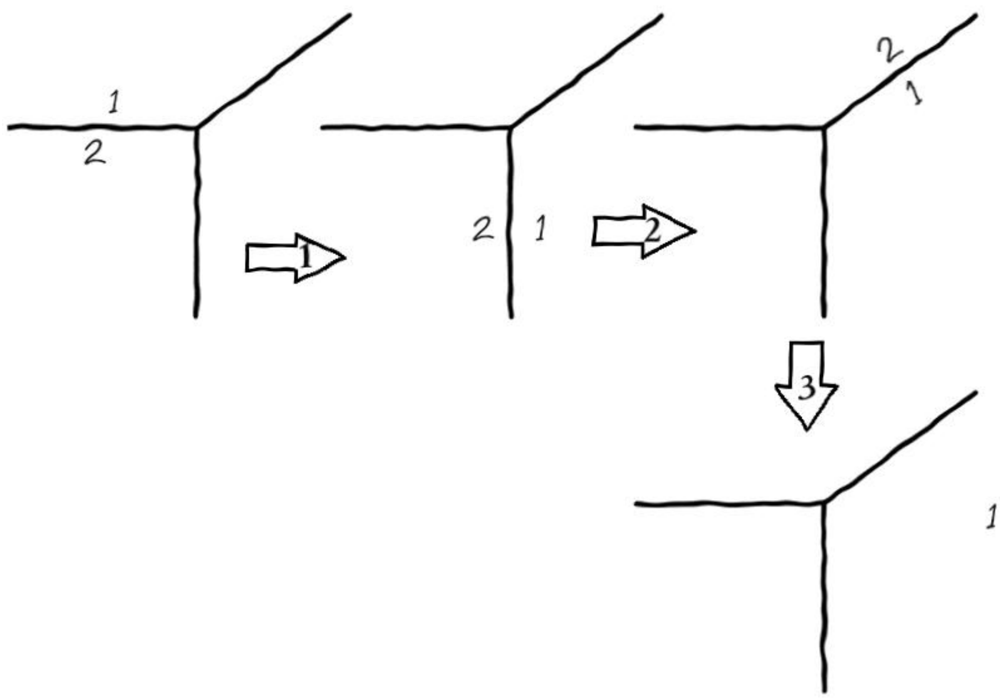


Figure 1.2: Examples of our new cube notation.

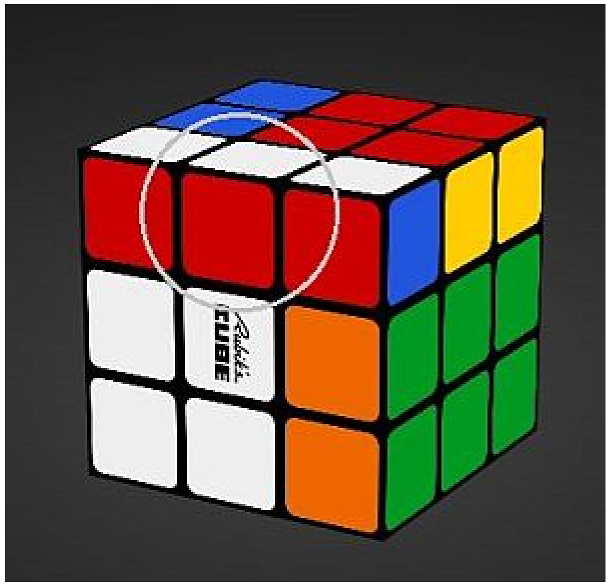


Figure 1.3: Edge flipped algorithm on a solved cube.

1.4 Sketch cube moves on paper

Our new cube notation allows you to draw cube moves by hand, so that an algorithm can be mentally "evaluated" on paper. To practice, let's *evaluate* a simple edge flip algorithm, shown in the earlier Figure 1.1.

The goal is to flip a $1_{up}2_{front}$ edge piece into a $2_{up}1_{front}$ edge. This algorithm only has three moves, as marked on the figure:

1. front face clockwise turn
2. right face clockwise turn
3. up face clockwise turn

As an exercise, please look at Figure 1.1 alone, and figure out which faces are turned and in which directions in sequence.

Now let's try this algorithm on a real cube. (Just remember that the trajectory of the edge piece **1**–**2** traces out a triangle in space.)

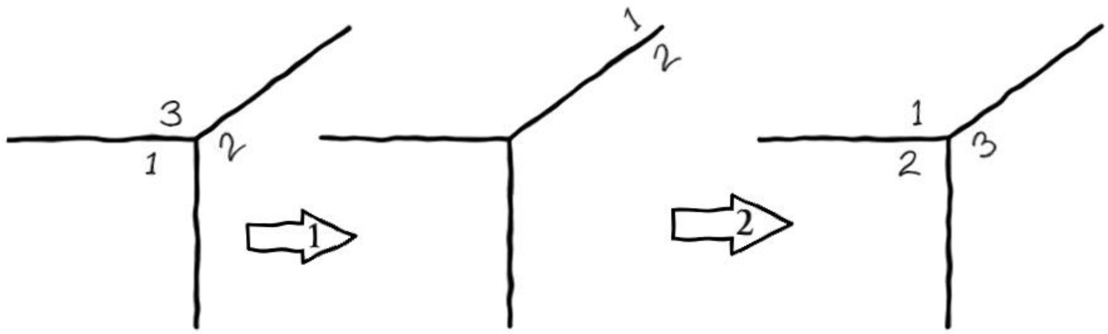


Figure 1.4: A simple corner twist algorithm.

Figure 1.3 shows the result of performing the edge flip algorithm on a solved cube. The white-red edge piece on the front face is indeed flipped. This algorithm works.

But you also notice that a lot of other pieces have moved out of place, that is scrambled, as well. Our sketch notation doesn't show these other pieces at all. Since these are not the intended effects of this algorithm, we call them "side effects".

Don't be discouraged. This is exactly our way of hiding the complexity of a problem, by focusing on one piece at a time. In the rest of the book, we will explore algorithms that would minimize these side effects, or use them to our advantage.

Our cube notation can track movements of edge pieces. It works equally well with corner pieces. Now let's see if we can twist a corner piece by sketching out an algorithm on paper.

This is even simpler, just 2 moves (Figure 1.4):

1. a right face clockwise turn; notice `2` stays on the right face, and `1` jumps an edge to the top face

2. an up face clockwise turn; notice `1` stays on the up face, and '2' jumps an edge to the front face

Figure 1.5 shows the result of the corner twist algorithm applied to a solved

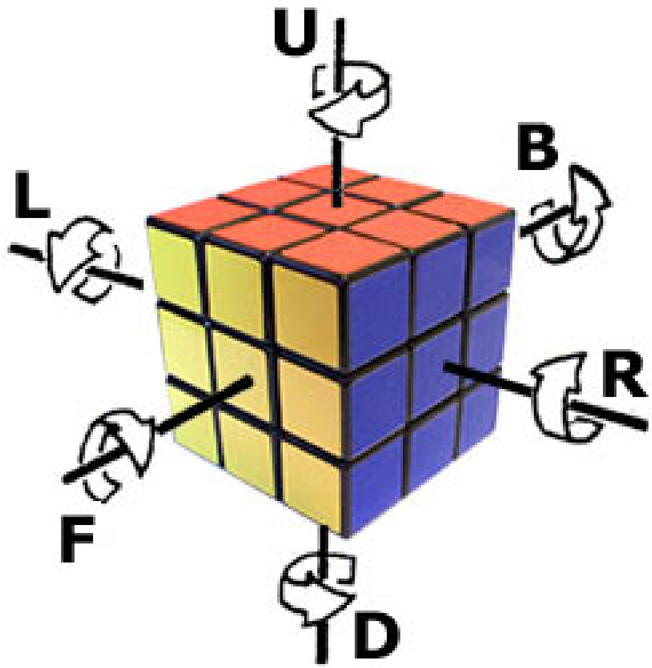
Figure 1.5: Corner twisted with a simple algorithm.

cube. A white–red–green corner is now twisted 120-degree into a green–white–red corner. Inadvertently other pieces of the cube are disturbed as well.

1.5 Traditional cube notation

In case you'd like to read other cube solving tutorial. Let's briefly mention the traditional cube notation.

The traditional cube notation assigns six letters to six faces of the cube. The six letters are: F(front), B(back), R(right), L(left), U(up), D(down).



Each letter is also an operator, representing a clockwise 90-degree turn of that specific face. For example, a clockwise 90-degree turn of the front face is denoted as F, a clockwise 180-degree turn as F^2. Similarly a counter-clockwise 90-degree turn becomes F^{-1}, following the mathematical concept of an "inverse".

$$F\,F^{-1} = I.$$

The above formula simply means a clockwise turn of the front face, followed by a counter-clockwise turn of the front face, cancels each other out.

Cube algorithms are spelled out with these face turn operators (being F, B, R, L, U, or D). Unless you can visualize the whole cube in your mind, looking at an algorithm itself doesn't help you understand **what's going on**.

So this traditional cube notation is a good documentation tool, but not a good tool of thought.

1.6 Summary

Our new cube notation tries to visualize the cube algorithms by sketching the result of each cube move, in contrary to the traditional cube notation, which explicit represents the cube moves symbolically or graphically.

Because humans are good at recognizing visual patterns and logic reasoning, but not memorization (at lease not without extensive training), our new cube notation helps you learn and understand cube algorithms much better.

Due to its simplicity, one can also “evaluate” cube algorithms on a piece of paper mentally.

Simple cube algorithms that flip an edge or twist a corner piece also wreak havoc on rest of the cube. We will discuss better algorithms with less side effects in the next chapter.

Reading about a cube algorithm on paper is a very different experience from turning a real physical cube. Hope you are curious and brave enough to put down this book and twist your hands on a real cube.

> We turn the cube and it twists us.
>
> —Erno Rubik

Chapter 2

Solving the first layer

We will explore the layer-by-layer method in this book. It is also called beginner's method. The scrambled cube is put back in order one layer at a time.

The advantage is that it's very easy to orient cube pieces and recognize cube patterns after the first layer is solved. The disadvantage is that once the first layer is solved, the subsequent algorithms will inevitably disturb that layer.

So the algorithms have to manage to: move pieces out of the way, perform some operation, and bring pieces back to their original places. That amounts to a lot of "extra" moves, comparing to other more speedy solving method.

2.1 Edge pieces on top layer

Pick a color for your top layer, for example *blue*. We are going to position 4 edge pieces first to form a cross on the *top* layer.

The two color squares of any edge piece are uniquely determined by the colors of the two faces that the edge crosses. In our example, if *yellow* is your front face, then the blue–yellow edge piece can only be in that location in Figure 2.1. The same goes for blue–red, blue–white, blue–orange edge pieces.

At this stage, you can make moves on almost all the faces without affecting anything else, since we haven't built much, except the top layer. Usually, we can use the bottom layer as our "conveyor belt" to move edge pieces around.

Once an edge piece **`1`**–**`2`** is in place on the bottom (Figure 2.2), it can be

Figure 2.1: Cross on the top layer.

moved to the top with a simple 180-degree turn. Keep in mind, as the 1–2 piece moves to the top layer, the original top layer edge drops down to the bottom layer, so it's like a *swap*. This *swap* concept can be quite useful later.

Up till now, we are assuming that the edge piece we like is already on the bottom layer, and it has the correct orientation. Otherwies, we will describe two algorithms. One moves an edge piece from the middle layer to the bottom layer. The other flips an edge piece to the correct orientation.

2.1.1 Case 1

Algorithm 1: Move an edge piece to the bottom layer.

Suppose 3–4 is already in place on the top layer, and you want to move the edge piece ⋆–⋆ to the bottom layer for transportation. Figure 2.3 shows the 3-step algorithm.

1. right face counter-clockwise turn—bring ⋆–⋆ to the bottom layer; note that 3–4 piece is now dragged out of place
2. move ⋆–⋆ out of the way
3. recover 3–4 edge back in place

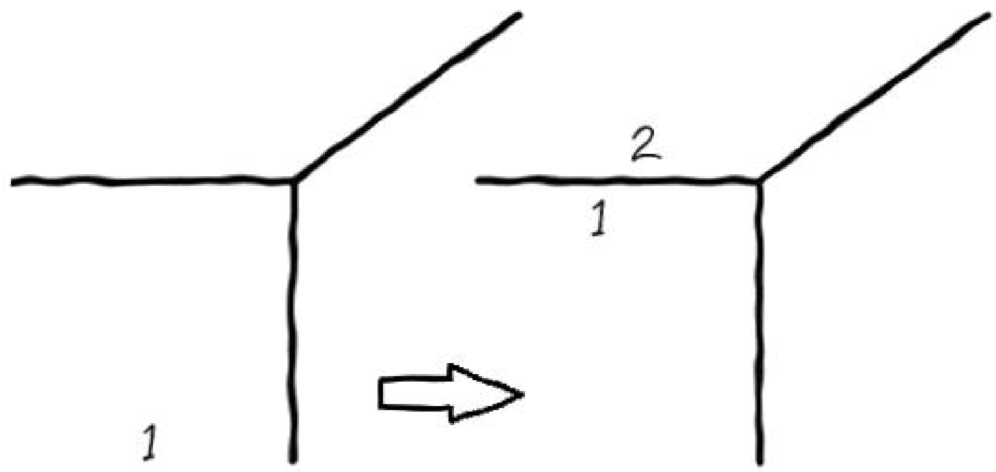


Figure 2.2: Edge piece swapped from bottom to top layer.

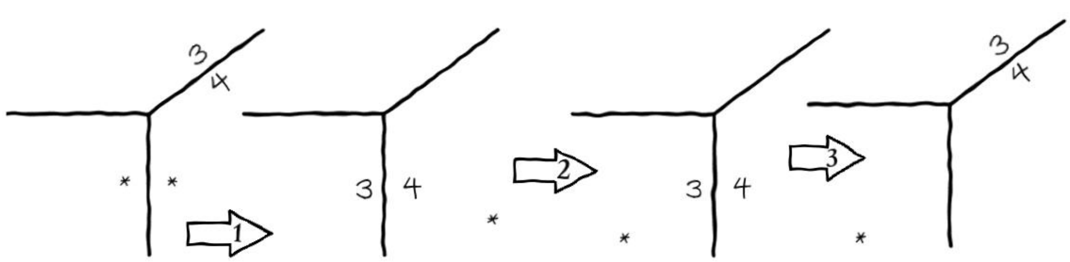


Figure 2.3: Move edge piece in the middle to the bottom.

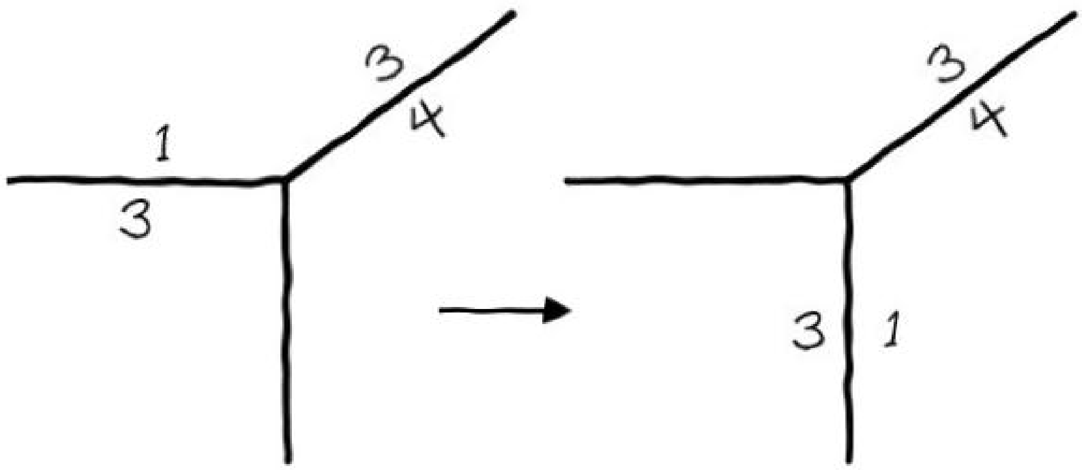


Figure 2.4: Edge piece in the wrong orientation.

Similarly *–* can also be brought down to the bottom layer via a clockwise turn of the front face., due to the symmetry of the right and front face. As an exercise, please sketch out the algorithm on paper.

2.1.2 Case 2

Now this time suppose the edge piece 3–4 is already in the right place, so is 1–3. Only that it needs to be flipped correctly, that is 3 on the up face and 1 on the front face.

Algorithm 2: Flip an edge piece.

A 90-degree clockwise turn of the front face positions 1–3 piece to be in the middle layer. That is exactly the start position for Algorithm 1 in Figure 2.3
.

Now we can use Algorithm 1 to move this 1–3 edge piece to the bottom layer. Next, we use a 180-degree turn to pull it back on top layer. Interest-

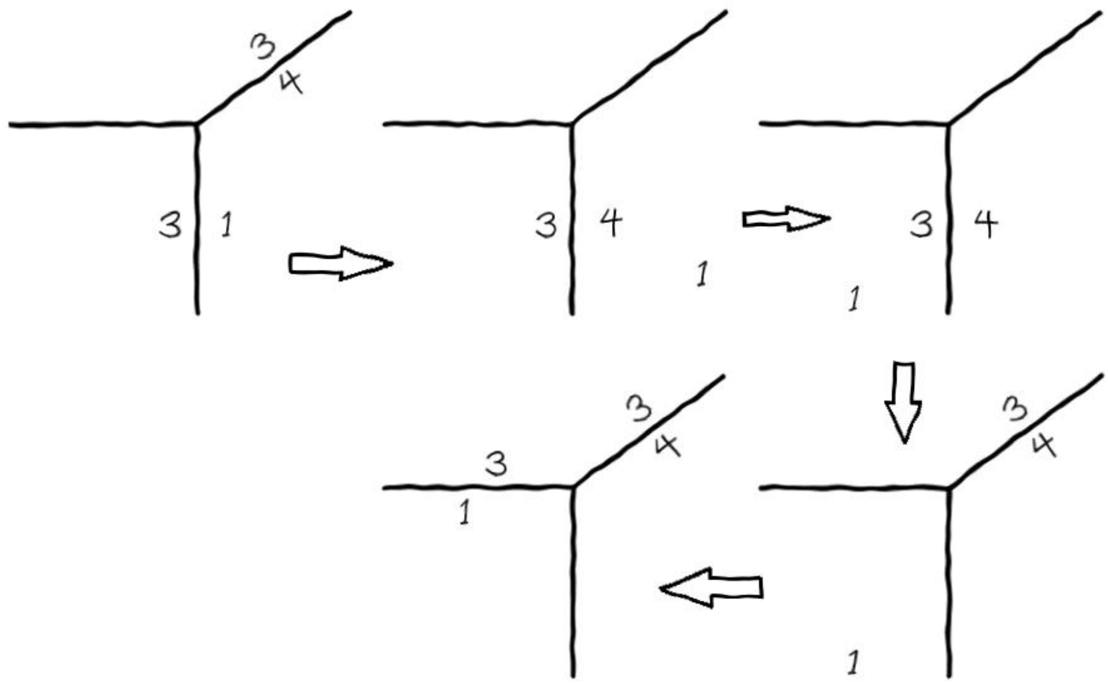


Figure 2.5: Edge flip algorithm.

ingly, the edge is now flipped correctly. This time we use the "side effect" of Algorithm 1 to our advantage.

Figure 2.5 shows the complete Algorithm 2 in the sketch notation. (As an exercise, please describe which face is turning in each step? BTW, the last step is a 180-degree turn.)

You might recall the earlier edge flip algorithm in Figure 1.1 . It is perfectly applicable if there is nothing else in place on the top layer. In fact to avoid disturbing the top layer, you can flip the cube upside down, to make the bottom layer your new top layer, then you can still use that algorithm to flip an edge. Just remember to recover any other edge pieces you may have knocked out of place along the way.

Figure 2.6: Finished first layer.

2.2 Swap corner pieces

The next step to complete the top layer is to put all four corner pieces in place, with the correct orientation (Figure 2.6).

Each corner piece has 3 colors, which are uniquely determined by the colors of the 3 faces it engulfs.

Again we will use the bottom layer as our conveyor belt to transport corner pieces to the right place. We assume the needed corner piece is already on the bottom layer. Otherwise we can use the *swap move* as shown in Figure 2.7 to swap a top layer corner to the bottom layer without breaking any pieces on the top layer.

This is (Figure 2.7) in essence similar to Algorithm 1.

Earlier, an edge piece can have 2 kinds of orientation, so there needs only one edge-flip algorithm. Now a corner piece can have 3 possible orientations.

2.2.1 Case 1

The corner piece is right under its intended position (Figure 2.8). Intuitively a clockwise turn of the right face will bring the corner **1**–**2**–**3** up, and fit perfectly in place. But that move also knocks the edge piece **1**–**3** out of place.

Let's explore a better algorithm (Figure 2.9) on paper. The idea is to bring the edge piece down to join the corner piece at the bottom, then bring them back together to the top layer.

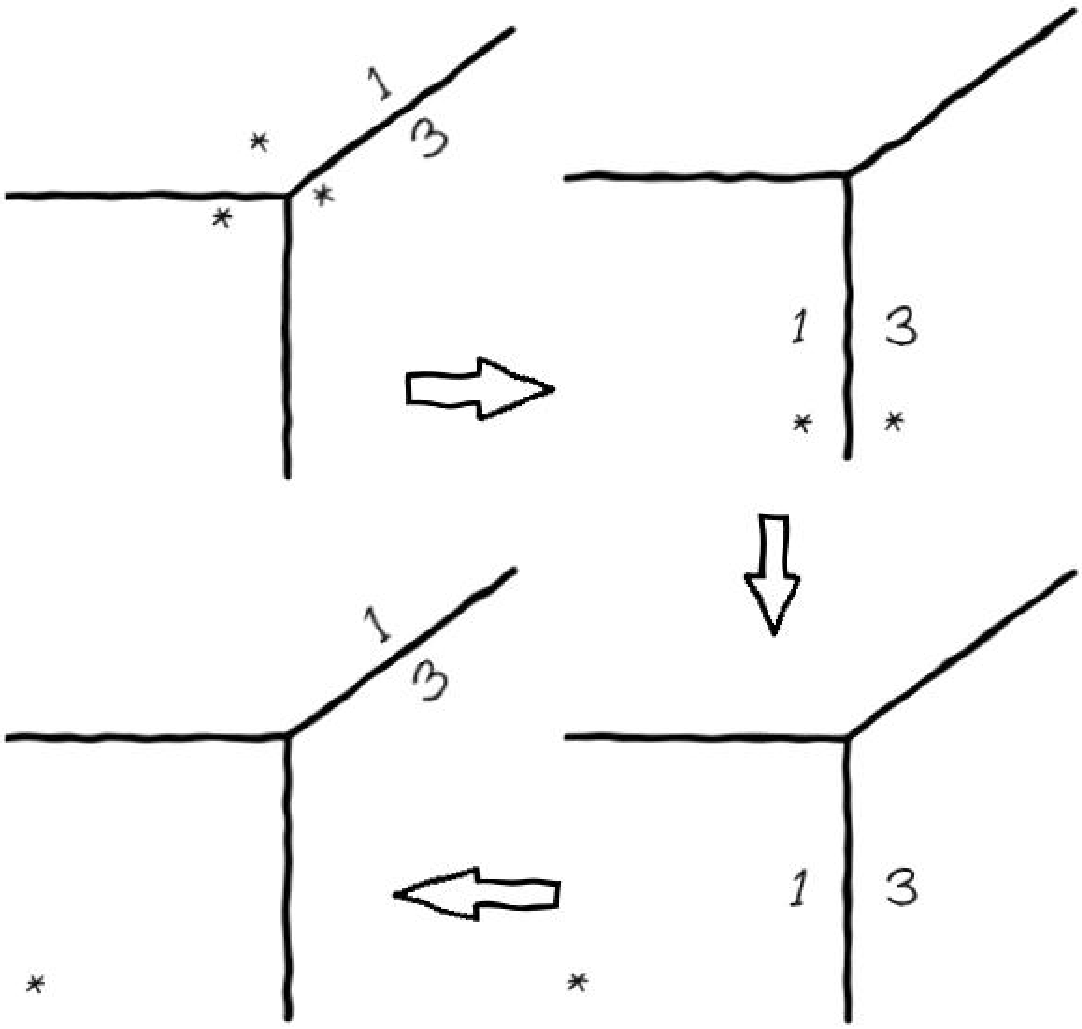


Figure 2.7: Swap top layer corner piece to bottom layer.

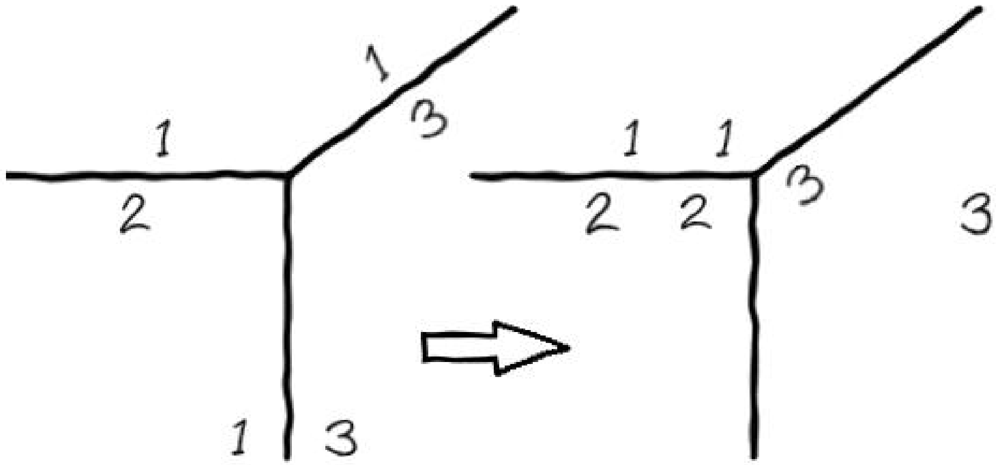


Figure 2.8: Intuitive move of the corner piece in place.

Algorithm 3: Swap a bottom layer corner pieces up via an *elevator*.

1. move the corner **1**–**2**–**3** out of the way
2. lower edge piece **1**–**3** down
3. join edge and corner pieces together
4. pull both pieces back up via right face clockwise turn

2.2.2 Case 2

As an exercise, you should be able to reason out how to solve the case in Figure 2.10. Hint: in this case, the corner piece is twisted 120-degree, now **1**–**2** edge piece needs to come down and join with the corner **2**–**3**–**1**, but before that the corner piece needs to move out of the way.

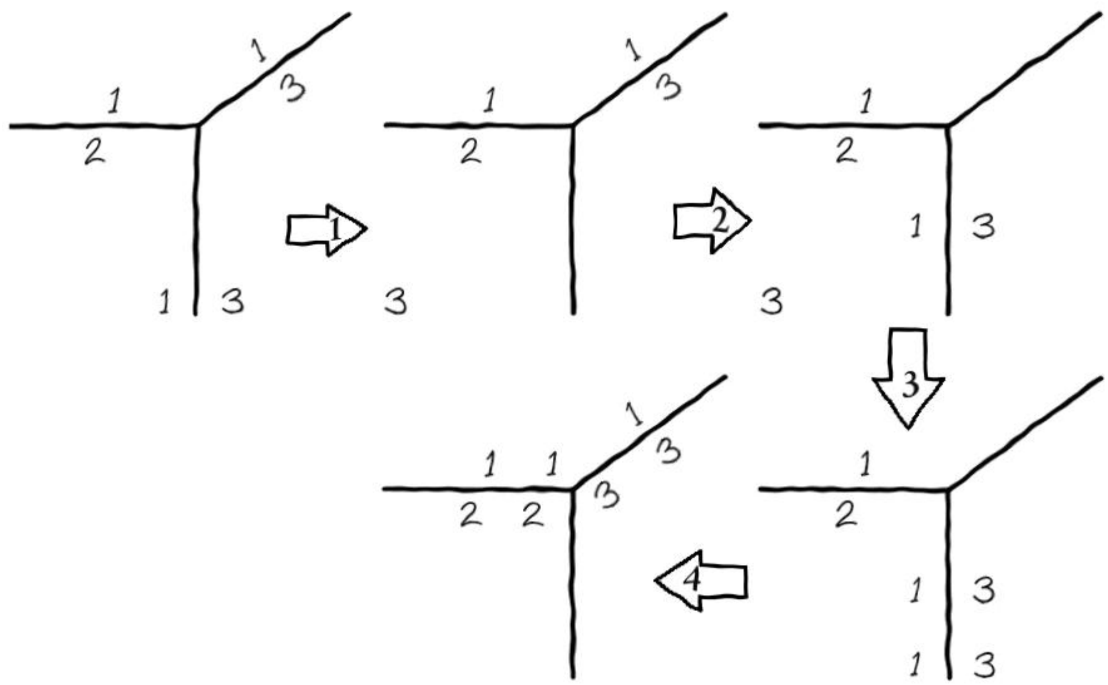


Figure 2.9: Swap a bottom layer corner pieces up via an elevator move.

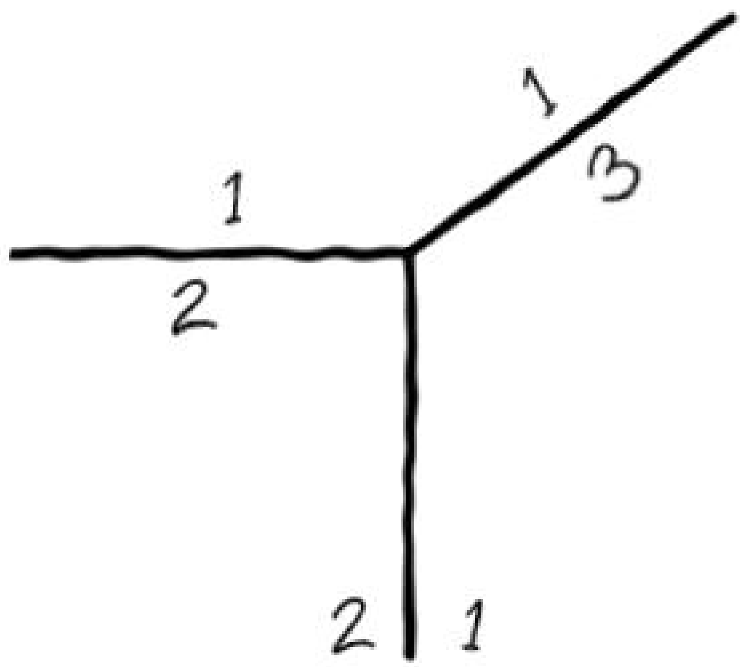


Figure 2.10: Move the bottom layer corner piece to the top.

2.2.3 Commutator and conjugate

If you are into cube theory, Algorithm 3 is also called a commutator, which obeys the formula $XYX^{-1}Y^{-1}$. In this case, X represents bottom-left turn, and Y represents right face counter-clockwise turn.

On the other hand, if you are the imaginative kind, you can image the pieces being cars at a STOP sign. The east-west bound cars are moving, interleaved by north-south bound cars. Algorithm 3 would be like: left–down–righ–up.

Some people also call this algorithm elevator, or chimney move, because a bottom piece moves up to the top layer via a right or left face turn at the end.

While we are at it, there is another common form of cube algorithm, called conjugate. It has the mathematical form XYX^{-1}. Figure 2.3 and Figure 2.7 are perfect examples of conjugates. The purpose of X and X^{-1} moves in a conjugate are to cancel out the side effects.

We've seen a commutator algorithm to bring a bottom layer corner piece to the top. Next we describe a conjugate algorithm to do the same thing. (Figure 2.11).

Algorithm 4: Swap corner piece via a conjugate.

1. front face clockwise turn; X
2. bottom layer right turn; joins the edge and corner pieces; Y
3. front face counter-clockwise turn ; X^{-1}

From the sketch notation above, we can clearly see that we can join a corner piece either with the right edge piece (Algorithm 3), or with the left edge piece (Algorithm 4). The former is a commutator, the latter is a conjugate.

2.2.4 Case 3

Because the corner piece can be twisted 120, 240, and 360 degrees, we have both Figure 2.9, 2.10, as well as 2.12 (left). In terms of Figure 2.12, the trick is

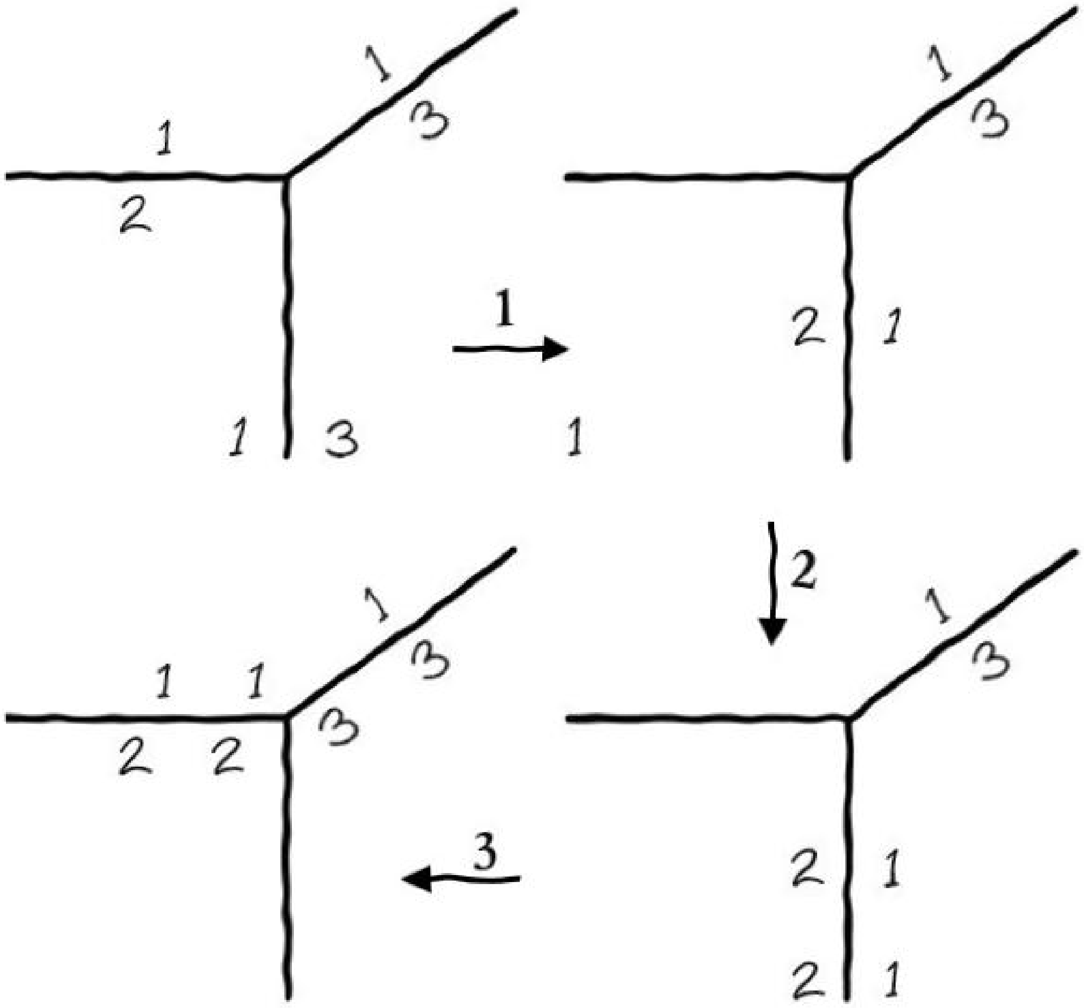


Figure 2.11: Swap corner piece via a conjugate.

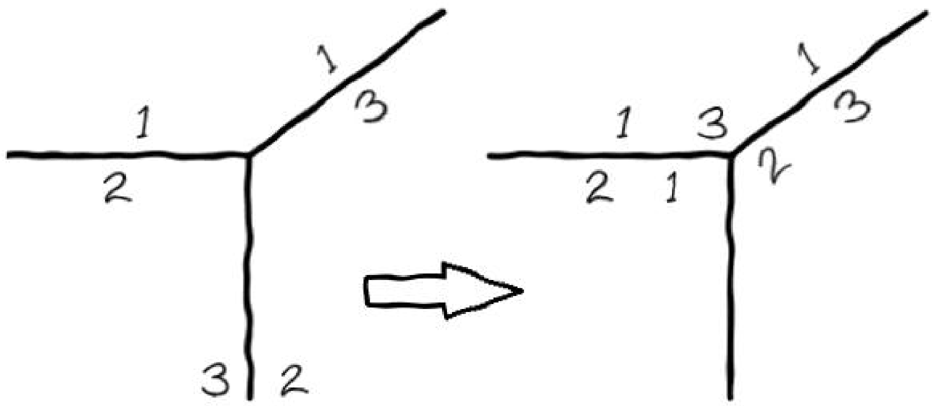


Figure 2.12: Corner piece join right face edge piece '1'–'3'.

to bring the corner up as before (Algorithm 3), of course, resulting in its wrong orientation (Figure 2.12 (right)).

Don't worry. The next step is to send the upper corner piece back to the bottom layer. This time, not via counter-clockwise turn of the right face, which will bury **1** down again, but via the front face clockwise turn. Just remember to recover the front face back with a conjugate move. Figure 2.13 shows how the end result looks like. Once the corner face **1** is at the front face, it can be solved just like before.

BTW, corner piece would never appear like that in Figure 2.14. Rubik's Cube just isn't made that way. It simply is not a valid permutation of the colors **1**, **2** and **3** for a corner piece.

2.3 Summary

I hope you are convinced that there is no need to memorize any algorithm to complete the first layer.

Once you work through the new cube notation in this book, you can developed a better understanding of the cube algorithms, and maybe start to discover your own algorithms on paper.

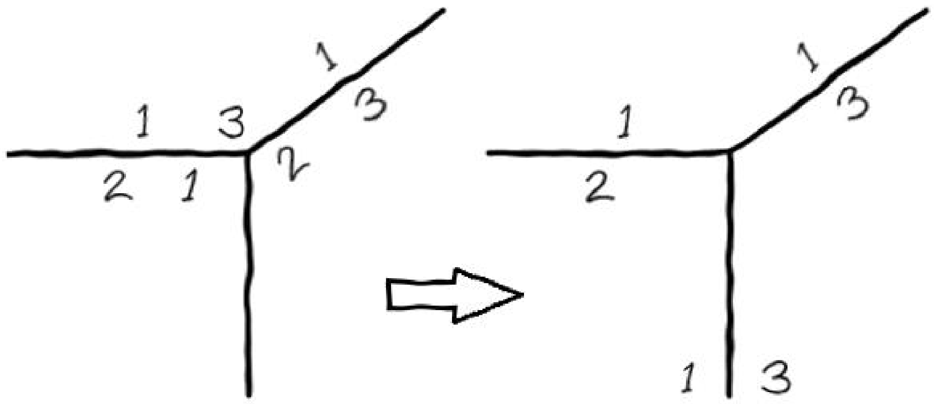


Figure 2.13: Swap corner piece via front face conjugate.

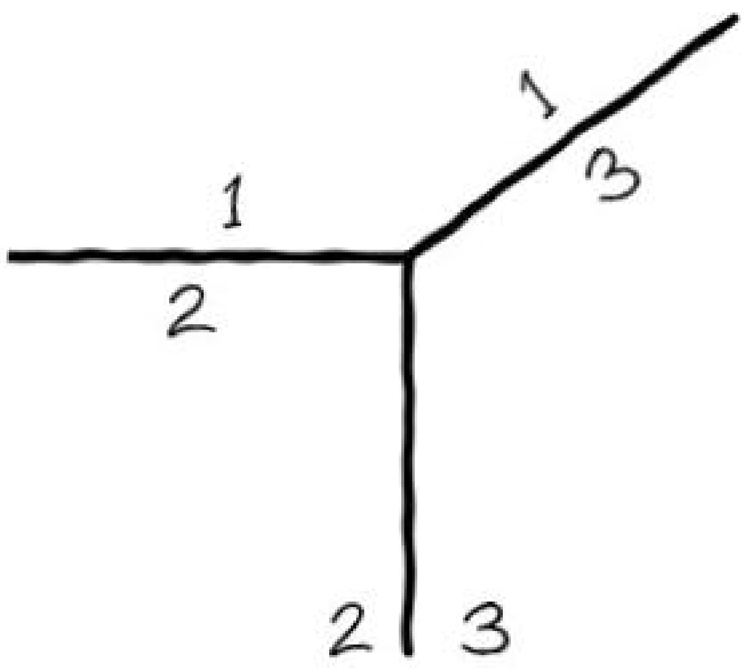


Figure 2.14: Corner configuration nonexistent.

For example, it would be a long sequence of letters in traditional cube notation, to spell out the algorithm solving Figure 2.12. But once you understand the logic of the algorithm, it is very easy to carried out the cube moves step-by-step through simple logic reasoning. You can even improvise your own algorithm along the way.

Admittedly, the first layer is the easiest to solve, because you can use the bottom layer to move pieces around. Any piece not on the bottom layer initially can be swapped into the bottom layer with simple moves, such as Algorithm 1.

Conjugate and commutator are two common structures of most cube algorithms. You don't have to be intimidate by their math formulas. Intuitively, they are easy to understand.

The symmetry of the front and the right face is worth noting as well.

Chapter 3

Solving the second layer

Now we have the top layer all solved, we can flip the whole cube upside down—the old bottom layer is now the new top layer. The objective is to fill the 4 edge pieces into the second layer. Of the remaining 8 edge pieces, 4 of them belong to the now top layer, the other 4 need to go into the second layer. In this case (Figure 3.1), suppose the top layer is in green, any edge piece that has a green square belongs to the top layer, and for example, a yellow–red edge piece on the top layer needs to come down into the second layer.

We will still use the top layer as our conveyor belt to move edge pieces around. But this time, any face turn, except the up face, will disturb the already solved bottom layer. We will describe a commutator algorithm that causes the least side effects.

The good news is that there are only 4 edge pieces to fit into the second layer, and only one algorithm to learn.

3.1 Join edge and corner pieces together

Suppose the **2**–**1** edge piece is on the top layer, ready to be moved to the second layer; in other words, to swap with piece ⋆–⋆ in Figure 3.2.

An intuitive approach would be like Figure 3.3.

1. bottom corner piece turns out of the way; X

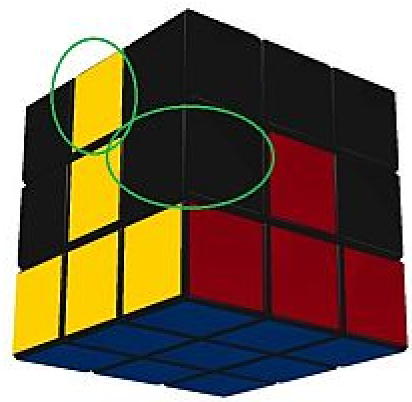


Figure 3.1: Move edge pieces from the top layer to the second layer.

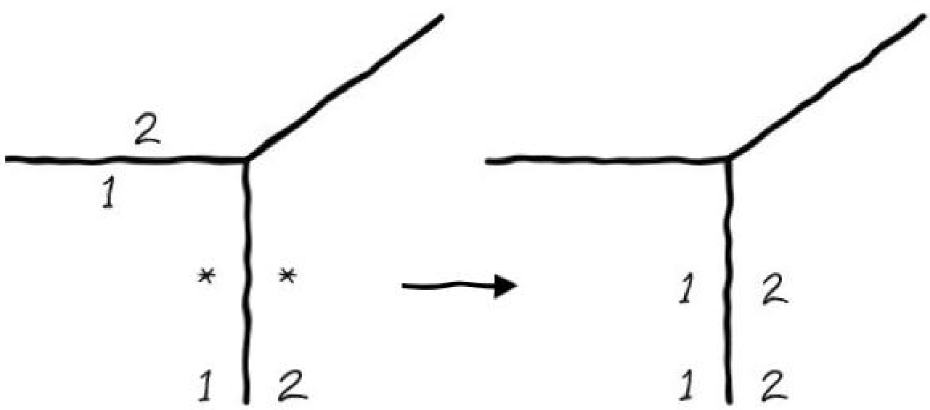


Figure 3.2: Swap edge pieces in the second layer.

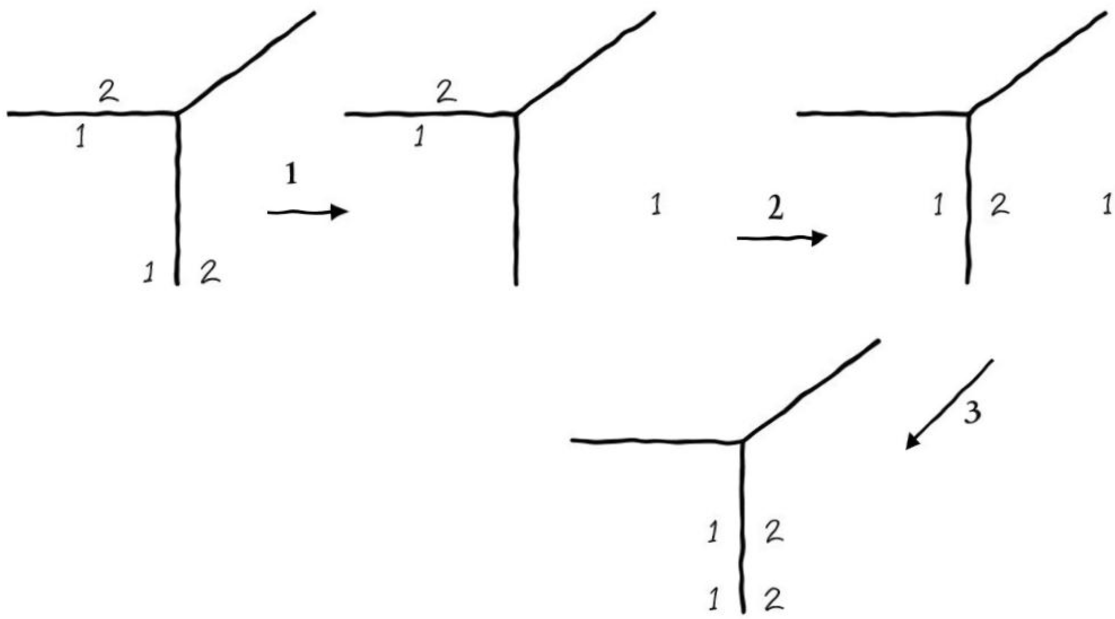


Figure 3.3: Move edge piece down the intuitive way (a conjugate).

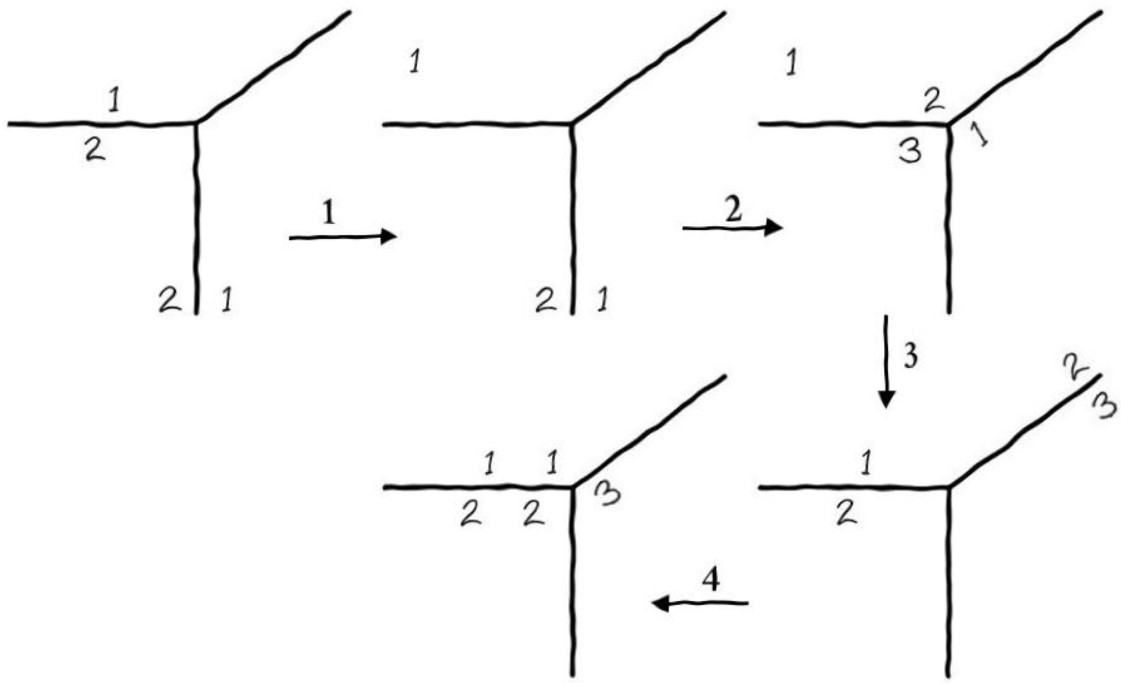


Figure 3.4: Join edge piece with corner piece.

2. top edge piece comes down; Y
3. bottom corner piece turns back; X^{-1}

But this approach disturbs the bottom layer, and makes it difficult to recover. As we learned in the previous chapter, a better strategy is to first join the edge piece with the corner piece, this time, **on the top layer** via a commutator.

Algorithm 5a: Join the edge piece with the corner piece on the top layer.

1. turn top layer edge piece away; X
2. move up corner piece; Y

3. turn the edge piece back; X^{-1}

4. join 2 pieces with right face counter-clocwise turn; Y^{-1}

This time, the side effect is minimal. The first step of Algorithm 5a is a little bit counter-intuitive. The nature of a commutator requires the edge piece to move away X , so that it can come back X^{-1} at step 3, ready for the "join".

The next step is to move these 2 pieces (**1–2** and **1–2–3**) down together. A front face clockwise turn would not work, since that disturbs the other pieces on the front face. The "elevator move" (another commutator) comes to help again.

3.2 Move edge and corner pieces together

Intuitively, we'd like to make a front face clockwise turn to move the **1–2** edge piece and **1–2–3** corner piece down together. That single move would disturb the bottom layer. A more useful way is to think of swapping out the **⋆–⋆** pieces, as in Figure 3.5.

Algorithm 5b: Move both corner and edge pieces down via the *elevator*.

1. turn **1–2** and **1–2–3** pieces away; X

2. front face **⋆–⋆** pieces up; Y

3. turn **1–2** and **1–2–3** pieces back; get into the elevator;X^{-1}

4. elevator these 2 pieces down; Y^{-1}

That completes Algorithm 5, which consists of these two a, b-stages.

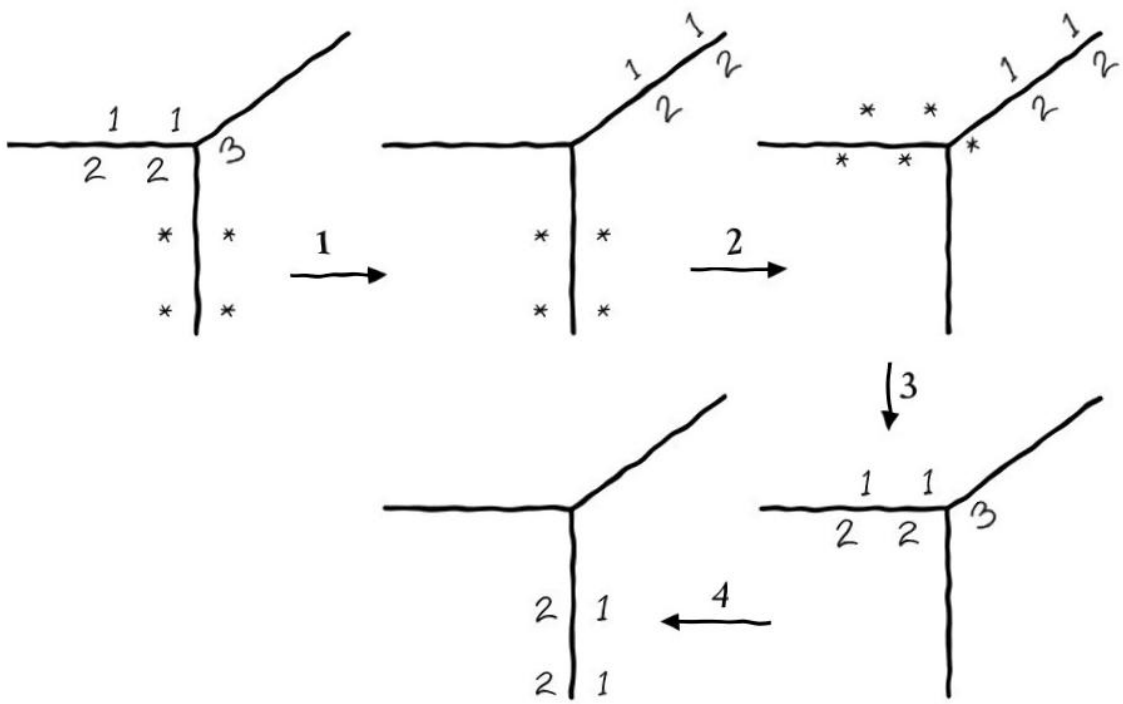


Figure 3.5: Move both edge and corner pieces down

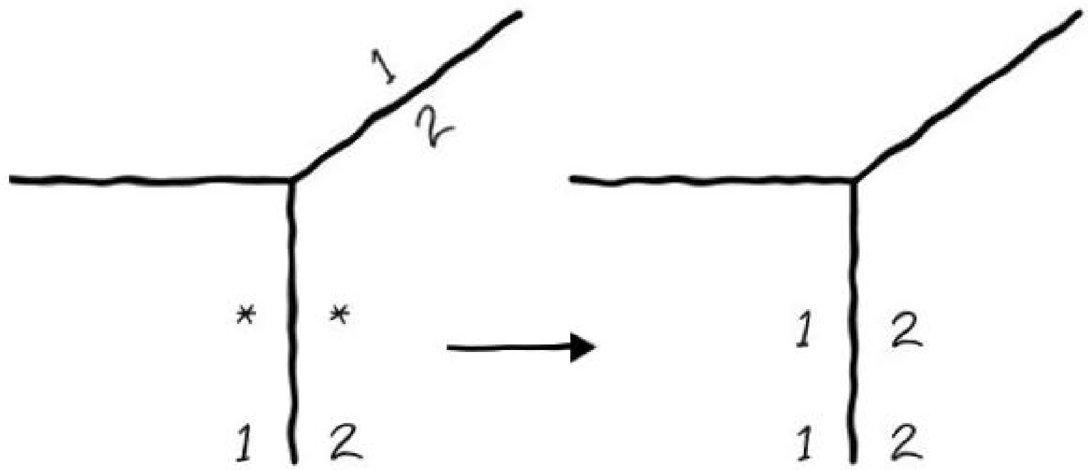


Figure 3.6: Top edge piece on the right.

3.3 Other cases

Algorithm 5 assumes that the needed edge piece already resides on the top layer; in case that edge piece is buried somewhere else on the second layer, you can still perform Algorithm 5 to *swap* the needed edge piece out to the top layer.

There is one more case we haven't discussed. Due to the symmetry of the front and right face, the edge piece on the top level can also be one the right face, as shown in Figure 3.6.

Its solution is left to you as an exercise. Please sketch out an algorithm on paper first and then try it on a real cube.

3.4 Summary

Algorithm 5 uses two commutators to solve the second layer without breaking the already solved first layer.

The key is to focus on the edge and corner pieces together: they need to be first joined on the top layer and then be moved down in one batch next.

With a good understanding of Algorithm 5, you should be able to devise your own way of solving Figure 3.6.

Chapter 4

Solving the final layer

There is only one layer left—4 edge pieces and 4 corner pieces to fit. The good news is that they are already on the top layer, so no need to transport or elevate them from other places. The bad news is that they still need to be rearranged, or sometimes called permutation.

Up to this point, you may find that the algorithms described in this book are almost identical to those in other beginner's method tutorials, except that we use a brand new cube notation to sketch out and explain the algorithms for the purpose of better understanding them.

In this final layer, the algorithms start to vary quite a lot, even though they all reach the same final goals. So don't worry if the algorithms described in this chapter don't agree with those in your cube booklet, or tutorial videos.

We didn't discover any of these algorithms ourselves. We made the effort to pick the ones that are simpler to execute and easier to understand. You can indeed solve the cube by following through this chapter to the end.

To solve the final layer, there are 4 algorithms for the following 4 steps:

1. flip the edge pieces
2. position the edge pieces
3. position the corner pieces
4. twist the corner pieces

4.1 Flip the edge pieces

Just like solving the first layer, we first build a cross by settling the 4 edge pieces on the final layer. Since all the edge pieces are already on this layer, the one problem is that some of them may have flipped the wrong way.

We can borrow the diagram in Figure 1.1 to show the problem: a $1_{up}2_{front}$ edge piece needs to flip into a $2_{up}1_{front}$ edge. In fact, we can use the exact algorithm described in section 1.4 as the starting point.

Algorithm 6a: Flip an edge piece (see Figure 1.1).

1. front face clockwise
2. right face clockwise
3. up face clockwise

By now the two bottom layers are in order, after performing Agorithm 6a, which successfully flips **1**–**2** edge, we can see the damage it causes. To recover, let's focus on the **bottom layer**.

It's easy to see that in oder to put the bottom layer back in order, the next logical step is:

Algorithm 6b: Recover the bottom layer.

1. right face counter-clockwise
2. up face counter-clockwise
3. front face counter-clockwise

So Algorithm 6 has two parts: one is to flip the edge as intended, the second part is to restore the bottom layer, by unwinding the turns of the first part. The order of the counter-clockwise turns are important. To clarify, the complete sequence, spelled out in traditional form is: $FRUR^{-1}U^{-1}F^{-1}$. But you don't

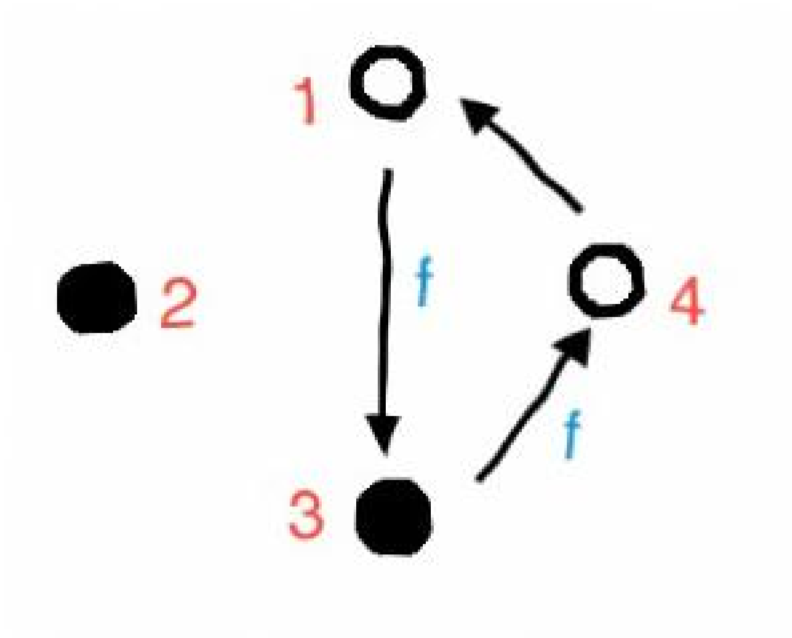


Figure 4.1: Edge pieces on the top layer (top view).

need to memorize this formula. Just remember to focus on the bottom layer during the second part of the algorithm.

Interestingly, there is more story to this algorithm. According to the cube theory, flipped edges are always in pairs. In other words, Algorithm 6 has other side effects. Figure 4.1 is a sketch, looking from the top at the top layer. 4 edge pieces are marked **1** to **4**. **3** is the edge piece along the front face, **4** is the edge piece along the right face, etc.

Figure 4.1 shows the trajectory of the 4 edge pieces after executing Algorithm 6. While edge piece **2** stays unchanged, **1** moves to **3**s place, **3** moves to **4**s place, and **4** moves to **1**s place. These 3 edge pieces simple shift around in a loop. Not only that, **1** gets flipped moving to position **3**, and **3** gets flipped moving to position **4**, but **4** keeps its orientation moving to position **1**. (Remember that cube theory: edge flippings always happen in pairs.)

Suppose we have the edge pieces top-view in Figure 4.2, where a black dot indicating a correctly flipped edge, a white dot indicating an incorrectly flipped

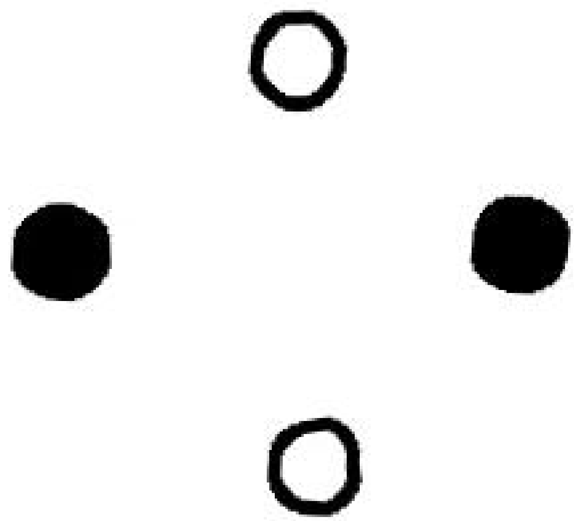
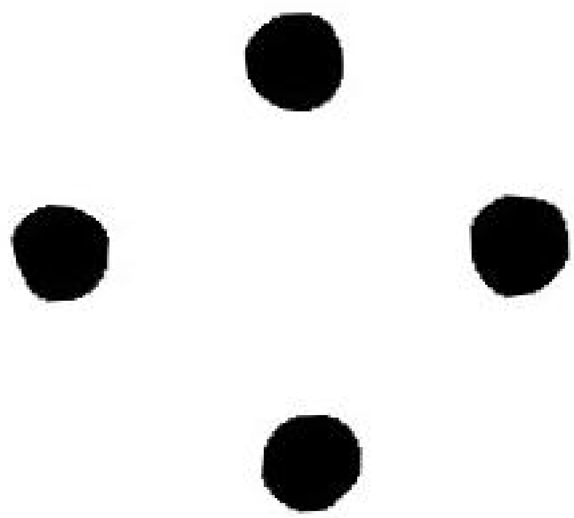


Figure 4.2: 2 edge pieces flipped correctly (black dots).

edge.

After executing Algorithm 6, according to Figure 4.1, they should all show up as correctly flipped.



So the trick is to put a correctly flipped edge along the left face, because it stays there unaffected by Algorithm 6. You may need to execute Algorithm 6 a few times. But once you have the pattern in Figure 4.2, one more iteration of Algorithm 6 will have all the edges flipped correctly.

Interestingly, according to Figure 4.1, if you have correctly flipped edges along the left and front faces, while the incorrectly flipped edges along the back and right faces, no matter how many times you execute Algorithm 6, they stay that way. (Please verify that for yourself.)

In order to use Algorithm 6 to flip all the edge pieces successfully, you need to look at the pattern of the existing edges on the top layer and choose the front face accordingly. For example, maybe none of the edge pieces are flipped correctly at the beginning. Then after running through Algorithm 6, there will be exactly 2 edge pieces flipped correctly. Now you need to choose which one of them to position on the left (top view).

Remember at this step, we only care about the top facing colors of the four edges. We will match the side colors in the next step.

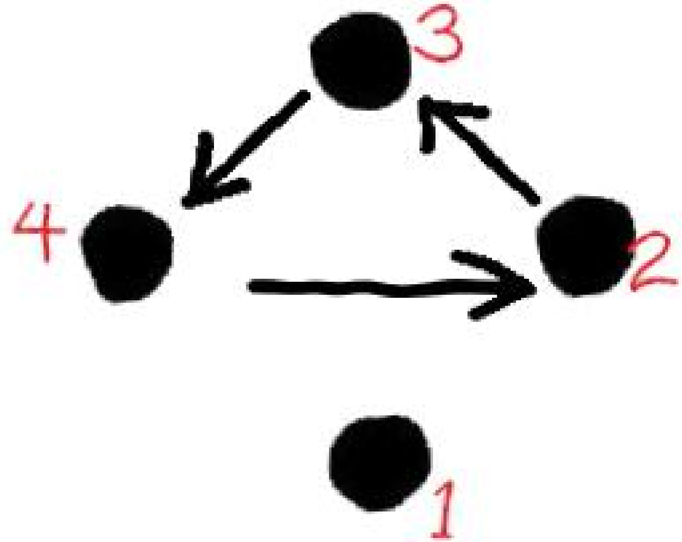


Figure 4.3: Edge permutation algorithm of the top layer (top view).

4.2 Permuting edge pieces

Once the edge pieces on the top layer all have the correct top facing color, we need to match the other color square of each edge piece to the color of each side of the cube.

We can rotate the up face to see if 4 edges match exactly the 4 sides, and 5 out of 6 chances that it doesn't happen automatically. In such cases, we need to move the edge pieces around, or called permuting the edge pieces.

The edge permutation algorithm (Algorithm 7) is probably the longest move in the beginner's method. Before we explain how to do it, let's see what it does.

Figure 4.3 is the top view of the top layer. Edge pieces are marked **1**–**4** in black dots. Unlike Algorithm 6, Algorithm 7 won't flip any edges this time. But it moves 3 edges in a loop:

1. front edge stays put

2. left edge jumps to right edge's place
3. right edge shifts to back edge's place
4. back edge shifts to left edge's place

Algorithm 7 has 7 turns. It is sketched out in Figure 4.4.

1. right face clockwise turn—bring ⋆–⋆ into top layer
2. up face clockwise turn—move ⋆–⋆ to the front face
3. right face counter-clockwise turn
4. up face clockwise turn—move ⋆–⋆ to the left face
5. right face clockwise turn–store **5**–**4** away
6. up face counter-clockwise turn 180-degree— bring ⋆–⋆ back to right face
7. right face counter-clockwise turn—bring ⋆–⋆ back down and **5**–**4** piece back up

First of all, all these right face and up face turns make sure the edge number **5** is always on top, so no edge is flipped. The ⋆–⋆ piece is injected into the top layer so that the edge piece **5**–**4** gets to store away on the right face at *step 5*. **5**–**4** piece comes back on to the top layer at *step 7* to complete the jump from the left to the right side.

When executing the algorithm, you can focus on the trajectory of the ⋆–⋆ edge piece.

Similar to Algorithm 6, you need to wisely decide your front face on executing Algorithm 7. Normally you would position the cube so that the already matched edge be your front edge.

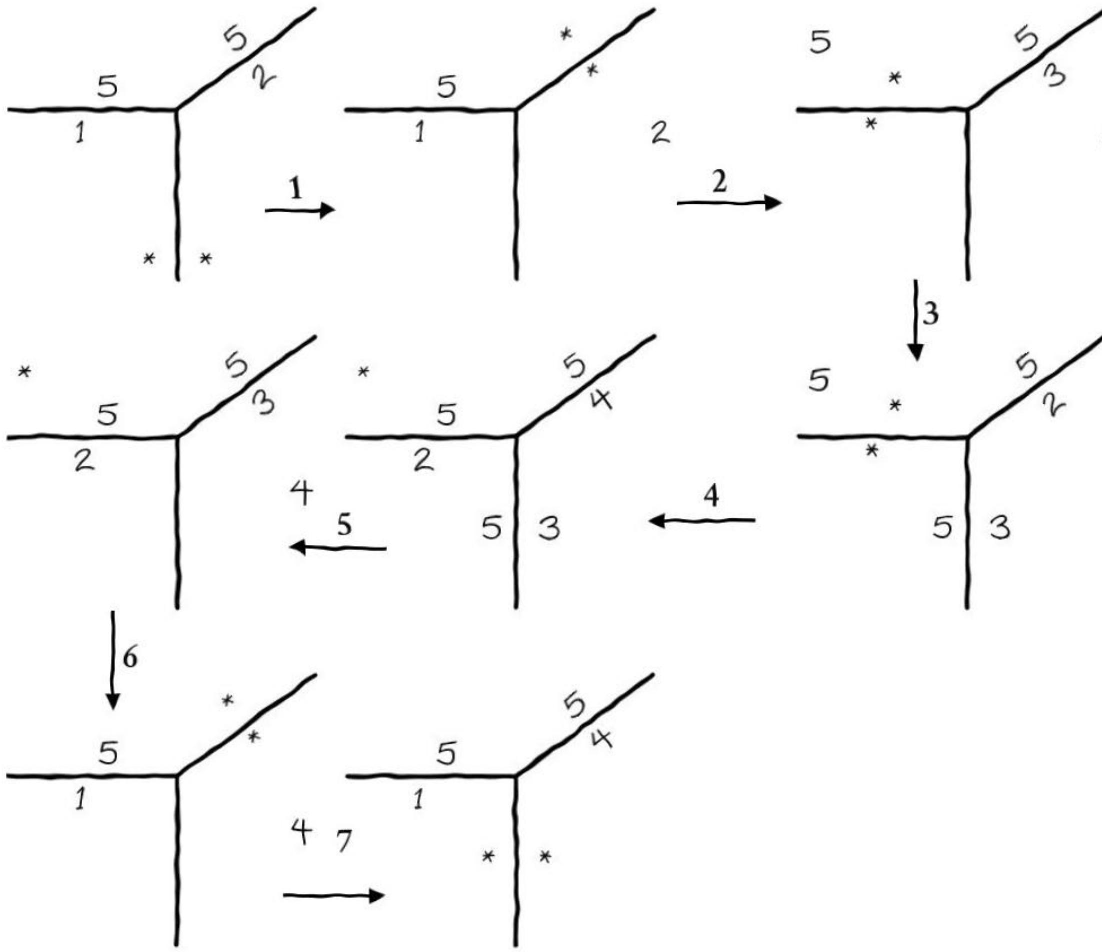


Figure 4.4: Edge permute algorithm.

4.3 Permuting corner pieces

Once the cross is solved on the top layer, we move on to the corner pieces. We position them correctly first in this section, then twist them to the right orientation in the next section.

Algorithm 8: Permuting corner pieces (Figure 4.5).

1. up face clockwise turn
2. right face clockwise turn—bring ⋆ corner into the mix
3. up face counter-clockwise turn
4. left face turn up—bring # corner into the mix
5. up face clockwise turn
6. right face counter-clockwise turn—remove ⋆ from the up face
7. up face counter-clockwise turn
8. left face turn down—remove # corner from the up face

As you can see, Algorithm 8 permutes the corner pieces B–C–D in a clockwise loop (Figure 4.5) and leaves corner piece A unchanged.

The strategy will be the same as before: find the already matched corner to be A, thus determine your front face. Then execute Algorithm 8, maybe multiple times, till 4 corner pieces are in the right place.

4.4 Twist corner pieces

A little cube theory before we dive into the final algorithm. Any cube algorithm that twist the orientation of a corner piece can do one of the following:

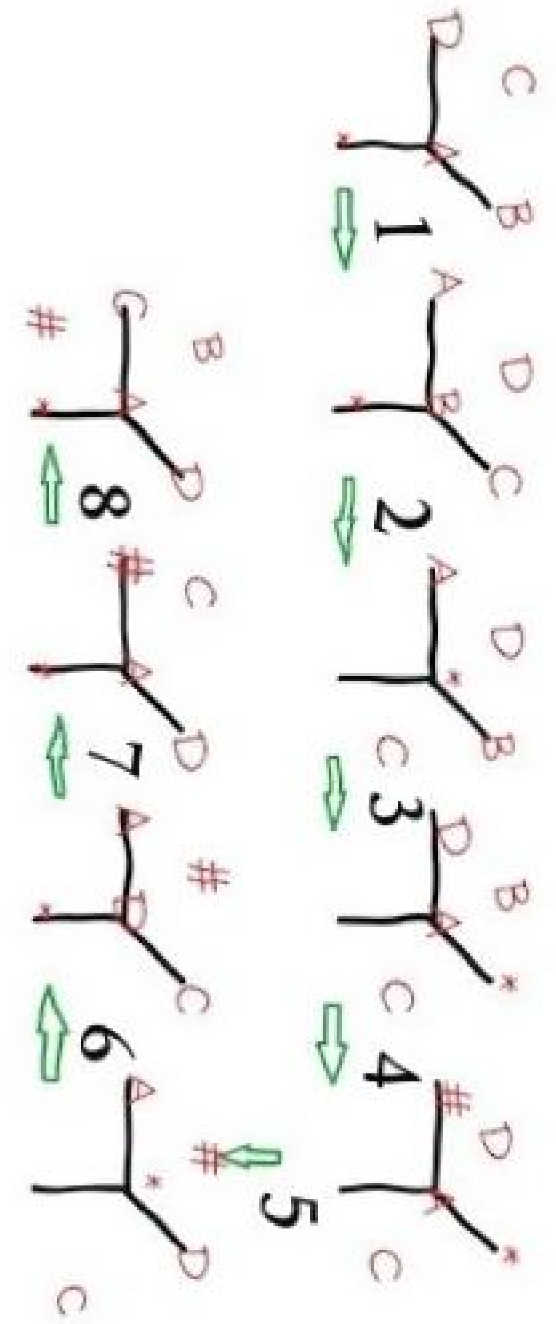


Figure 4.5: Corner permutation algorithm .

1. twist 3 different corner pieces, each 120-degree clockwise
2. twist any pair of corner pieces, in each pair, one corner with 120-degree clockwise, the other corner with 120-degree counter-clockwise

You get the picture. Just like we mentioned before, edge piece flips in pairs, but corner pieces have 3 different orientations. In this case, if you add up all the corner piece twists, it amounts to a constant 0-degree, or 360-degree. In other words, if you see a corner twisted 120-degree clockwise, there must be some other corner pieces somewhere on the cube to balance out that twist, by a 120-degree counter-clockwise turn, for example.

This last step uses a short commutator algorithm repeatedly (2, 4, 8, … times).

Algorithm 9: Twist corner pieces. (Figure 4.6)

1. right face down
2. bottom face left
3. right face up
4. bottom face right (repeat step 1 to 4)

Step 1–4 is a simple commutator. From the sketches, we can see that Algorithm 9 swaps two corners: **1–3–2** and **5–4–6**. It also twists each corner 120-degree. According to the cube theory, some other corners must be twisted to balance the act out. In fact, if you try this algorithm out on a real cube, you would be surprised that it wreak havoc on the whole cube. Lots of pieces gets moved around.

This time, with the leap of faith in cube theory, you do have to "blindly" follow this algorithm to the end.The idea is to pick an incorrectly twisted corner. Run Algorithm 9 from step 1 to step 8. If the corner is twisted correctly, great, turn the up face so that another incorrectly twisted corner is right on top of **4–5–6** corner. Now we apply Algorithm 9 again (maybe multiple times)

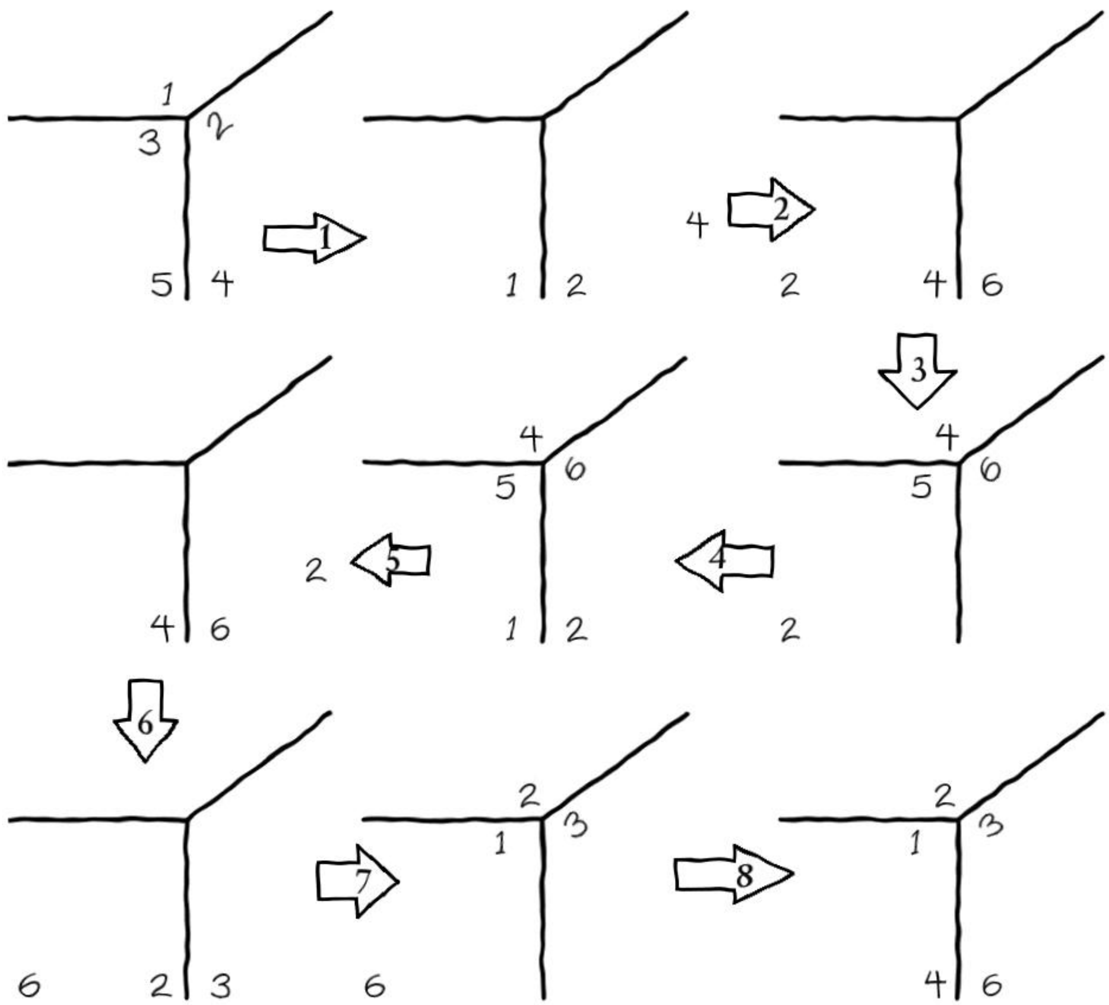


Figure 4.6: Corner twist algorithm.

until this other corner on the top level is twisted correct as well. In the end, all the corners on the top layer will be in the correct orientation. According to the cube theory, the **4**–**5**–**6** corner will be in the right orientation as well.

Common gotcha for this final algorithm:

1. panic, because all the other pieces are out of place temporarily; Please have faith in the cube theory

2. forget to execute "step 4" after the up corner piece is twisted correctly; Please remember it's a commutator, and **4**–**5**–**6** corner must be in the bottom layer position.

3. After one corner is twisted correctly, you rotate the whole cube to a different position and start Algorithm 9 from scratch; Please don't leave corner **4**–**5**–**6** hanging there with the wrong orientation. All the incorrectly twisted corner needs to be twisted at the position on top of **4**–**5**–**6** for Algorithm 9 to work its magic.

4. Give up after Algorithm 9 fails to work. Please have patience and try again. Maybe you missed a step somewhere in between. Even if you solve the cube for one time, that will be a great confidence boost. You've reached this far, don't give up easily.