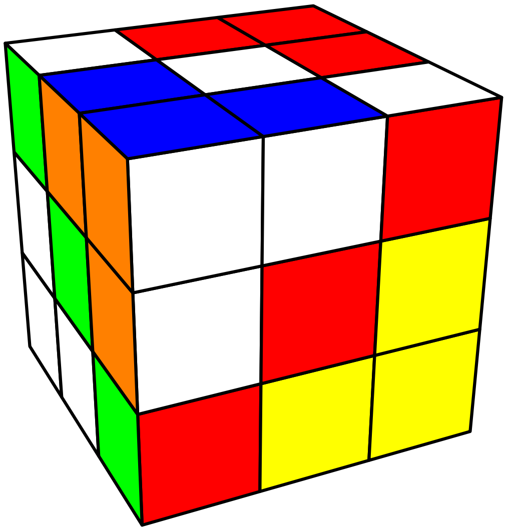
Group Theory via Rubik's Cube



Tom Davis
tomrdavis@earthlink.net
http://www.geometer.org

ROUGH DRAFT!!!

December 6, 2006

Abstract

A group is a mathematical object of great importance, but the usual study of group theory is highly abstract and therefore difficult for many students to understand. A very important class of groups are so-called permutation groups which are very closely related to Rubik's cube. Thus, in addition to being a fiendishly difficult puzzle, Rubik's cube provides many concrete examples of groups and of applications of group theory.

In this document, we'll alternate between a study of group theory and of Rubik's cube, using group theory to find tools to solve the cube and using the cube to illustrate many of the important topics in group theory.

1 Introduction

Note: If you have a new physical cube, do not jumble it up right away. There are some exercises at the beginning of Section 2 that are much easier with a solved cube. If you have jumbled it already, it's not a big deal—Appendix A explains how to unjumble it but the first few times you try, you'll probably make a mistake.

To read this paper you will certainly need to have the **Rubik** computer program and it would be very good also to have a physical Rubik's cube. The **Rubik** program, complete documentation for it, and a few sample control files may be obtained free of charge for either Windows or Mac OS X (version 10.2.0 or later) at:

`www.geometer.org/rubik`.

If you have not done so, acquire a copy of the program and print a copy of the documentation (there's not too much—only about 15 pages). If you don't have **Rubik**, but do have a cube, you'll need a lot of patience and probably a screwdriver to take the cube apart for reassembly in a "solved" configuration if you don't know how to solve it already.

First, some quick notation. The word "cube" will usually refer to the entire cube that appears to be divided into 27 smaller cubes. We shall call these smaller cubes "cubies", of which 26 are visible. There are three types of cubies: some show only one face (called "face cubies" or "center cubies", some show two faces, called "edge cubies" ("edgies"?) and some show three: the "corner cubies" ("cornies"?). The entire cube has six faces, each of which is divided into 9 smaller faces of the individual cubies. When it is important to distinguish between the faces of the large cube and the little faces on the cubies, we'll call the little faces "facelets".

A permutation is a rearrangement of things. If you consider the "things" to be the facelets on Rubik's cube, it is clear that every twist of a face is a rearrangement of those facelets. Obviously, in Rubik's cube there are constraints on what rearrangements are possible, but that is part of what makes it so interesting. The three facelets that appear on a particular corner cubie, for example, will remain next to each other in every possible rearrangement.

A good understanding of permutations and how they behave will help you to learn to effectively manipulate and solve Rubik's cube. The cube, however, has 54 visible facelets, so each cube movement effectively rearranges many of the 54 items. The best way to learn about any mathematical subject is to begin by looking at smaller, simpler cases. Thus in the first part of this document we'll look at permutations of small numbers of items, where we can list all the possibilities and easily keep everything in mind.

When we talk about general properties of permutations in the following text, try to think about what these statements mean in the context of a few concrete examples. Rubik's cube is one such concrete example, and we'll introduce a few others as we proceed.

2 The Rubik Program and the Physical Cube

If your physical cube is solved (as it came when you bought it), continue with the following exercises. If it is jumbled, get it unjumbled first by following the directions in Appendix A and then return here. And if you make a mistake while reading this section and accidentally jumble your cube so that you can't solve it, you'll probably need to do the same thing. In fact, even if you've got a solved cube now, it is almost certain that you'll make a mistake sometime as you read, so it's a good idea to try out the method in the appendix to make sure you know how it works. Take your solved cube and make one or two twists, then make sure you can use the **Rubik** program to find that one- or two-move solution.

Beginning now and for the rest of the paper, we will use the same notation to describe the cubies and the twists that is used by the **Rubik** program. For complete details, see the **Rubik** documentation in the section entitled, "Cube Coordinates and Move Descriptions".

Basically, what you'll need to know now is that the letters: U, L, F, R, B and D correspond to quarter-turn clockwise twists about the up, left, front, right, back and down faces, respectively. "Clockwise" refers to the direction to turn the face if you are looking directly at the face. Thus if you hold the cube looking at the front face, the move B appears to turn the back face counter-clockwise. The lower-case versions of those letters, u, l, et cetera, refer to quarter-turn counter-clockwise moves about the respective faces.

Hint: if you are beginning, it might be a good idea to put temporary stickers on the six center facelets of your physical cube labeled "U", "L", et cetera, and then just make certain that your cube has the same up and right faces as the virtual cube on the computer screen if you wish to use the two in conjunction (like when you're using **Rubik** to unjumble your physical cube). At the very least, decide for yourself on a "standard" orientation, like "white face up, green face left" (which happens to be **Rubik**'s default orientation). With these temporary labels in place you can't use the whole-cube moves or the slice moves since they may change which cubie is "up" or "left".

2.1 Inverse Operations

Let's begin with a couple of obvious observations. If you grab the front face and give it a quarter-turn clockwise (in other words, you apply an F move), you can undo that by turning the same face a quarter-turn counter-clockwise (by doing a f move). If you do a more complicated operation, like F followed by R, you can undo that with a r followed by a f. *Notice that you need to reverse the order of the moves you undo in addition to the direction of the turns—if you try to undo your* FR *sequence with an* fr *you will not return to a solved cube.* Try it—carefully do the sequence FRfr and note that the cube is not solved.

If you have just applied the above sequence, FRfr, to return to solved, you'll need to do a RFrf. Do you see why? Do so now to return your cube to "solved".

In mathematics, an operation that "undoes" a particular operation is called the inverse of that particular operation, and the inverse is often indicated with a little "-1" as an exponent. If we wanted to use this convention with our cube notation, we could write "F^{-1}" in place of "f", "U^{-1}" instead of "u" and so on. Since the standard computer keyboard does not allow you to type exponents, the lower-case versus upper-case notation is not only easier to type, but is more convenient. Keep in mind, however, that if you read almost any mathematical text that works with operations and their inverses, the use of "-1" as an exponenet is the usual way to express an inverse.

This double-reversal idea (that RFrf is the inverse of FRfr is very general. If a, b, c, ... are *any* operations that have inverses a^{-1}, b^{-1}, c^{-1} and so on, then:

$$(abc\cdots xyz)^{-1} = z^{-1}y^{-1}x^{-1}\cdots c^{-1}b^{-1}a^{-1}.$$

Because of this general principle, it is thus trivial to write down the inverse of a sequence of cube moves: just reverse the list and then change the case of each letter from upper to lower or vice-versa. For example, the inverse of the sequence ffRuDlU is uLdUrFF. This will always work.

Notice also that in the case of Rubik's cube moves another way to write the inverse of F is as FFF. In other words, if you twist the front face three more times, that's the same as undoing the original twist. We'll look more at this idea in the following section.

In the paragraph above, we consider a situation where we apply the same operation (F) three times in a row. This is a very common thing to do, and some operations will be applied many more than three times in a row. For this reason, we use an exponential notation to indicate that an operation is repeated. Thus for "FFF" we will write "F ^{3}" where the "3" in the exponent indicates that the operation is repeated three times. The exponential notation can be used to apply to a group of operations. For example, if we want to do the following operation: FRFRFRFRFR (in other words, apply the combined FR operation five times), we can indicate it as follows: $(\mathsf{FR})^5$.

To make sure you understand this notation, is it clear that $\mathsf{F}^9 = \mathsf{F}$? That's because the operation F applied four times in a row returns the cube to the state before you started. Although it may not seem important now, it *is* very important to have a name for the operation of "doing nothing". We call this the identity operation and will label it here as "1". See Section 3.1. (You can think of the "1" in terms of multiplication: multiplication by 1 in our usual number system effectively does nothing.

3 Commutativity and Non-Commutativity

Again it should be obvious, but the order in which you apply twists to the faces makes a difference. Take your physical cube and apply an FR to it and apply RF to the virtual cube in **Rubik**. It's obvious that the results are different. Thus, in general FR $\neq$ RF. This is *not* like what you are used to in ordinary arithmetic where if you multiply two numbers together, the order doesn't matter—$7 \times 9 = 9 \times 7$ and there's nothing special about 7 and 9.

When the order does not matter, as in multiplication of numbers, we call the operation "commutative". If it does matter, as in the application of twists to a cube, or for division of numbers ($7/3 \neq 3/7$) then we say that the operation is non-commutative. It's easy to remember the name; you know what a commuter is: someone who commutes, or moves. If an operation is commutative, the objects can commute across the operation and the order doesn't matter.

Just because a system is non-commutative, that does not mean that the result is always different when you reverse the order. In your cube, for example, FB $=$ BF, UD $=$ DU and LR $=$ RL, $\mathsf{FF}^2 = \mathsf{F}^2\mathsf{F}$, and so on. (And in arithmetic, division is sometimes commutative: $1/(-1)$=$(-1)/1$.)

If twisting the cube faces were a commutative operation, then solving the cube would be trivial. You would just need to make sure that the total number of F turns, U turns, and so on, are multiples of 4 and you would be done. To see this on a small scale, suppose your cube only allowed you to turn the front and back faces but turns about the left, right, up and down faces were not allowed. Try this with your physical cube, and you'll see that it's not a very interesting puzzle.

3.1 Order

Since we are looking at *all* operations that can be performed on a cube, it is important that we not forget perhaps the most important one: the operation of doing nothing—of leaving the cube exactly as it was before. This is called the "identity" operation and we'll call it 1 here if we need to refer to it. The reason that 1 is a reasonable notation is that if we use the notation FRB to mean F followed by R followed by B, it sort of looks like we're "multiplying" together those three operations. We're also used to the idea that multiplying anything by 1 leaves it unchanged, and it's certainly true that $\mathsf{1F} = \mathsf{F1} = \mathsf{F}$—doing nothing and then doing F is the same as just doing F.

Let's begin by looking at another obvious thing. If you start with a solved cube and perform the R operation on it four times, the resulting cube returns to a solved state. Since our notation for combining moves makes us think of multiplication (and as we shall see, this is a good way to think of it), we could indicate multiples of the same operation as exponents: $\mathsf{FF} = \mathsf{F}^2$, $\mathsf{FFF} = \mathsf{F}^3$, et cetera. Now, since we noticed that applying the f operation four times was the same as doing nothing, we can also write F^4=1.

As we do in most other areas of mathematics, it is reasonable to define $\mathsf{F}^0 = \mathsf{1}$, since applying an operation zero times is the same as not applying it at all, which is our definition of 1. Similarly, $\mathsf{F}^1 = \mathsf{F}$ since an exponent of 1 corresponds to applying the operation once.

Obviously, there is nothing special about F for this exponential notation—it applies to any other move, or, in fact, to any combination of moves. For example, if we thing of the combination FR as a single operation, then if we want notation that corresponds to repeating that operation 5 times, we can write $(\mathsf{FR})^5$. This means exactly the same thing as FRFRFRFRFR.

In this case it is also obvious that if we look at successive powers of F: F^1, F^2, F^3, and so on, then F^4 is the first time that we return to the identity. For this reason, we say that the "order" of the operation F is 4; four moves do the job and no smaller number of moves return us to where we started.

What appears at first to be somewhat amazing is that *any* cube operation has such an order. In other words, if you begin with a solved cube and repeat any operation enough times, the cube will eventually return to "solved".

As an exercise, try to find the order of FFRR using a physical cube. Start with a solved cube and apply those four moves. You will find that the cube is a bit jumbled. Repeat the same four moves, and again, and again. Eventually (assuming you don't make a mistake), the cube will be solved again. The total number of times you had to repeat the four-move combination is the order of that operation. Do try this. You will not have to repeat the operation too many times.

You can check your answer with **Rubik**. Reset the cube to solved and type “FFRR” into the window labeled “Current Macro”. Then press the $\boxed{\text{Macro Order}}$ button just above the window in which you just typed, and **Rubik** will pop up an information window showing you the order that it calculated. Or, if you don’t trust the $\boxed{\text{Macro Order}}$ button’s result, you can apply the FFRR operation repeatedly until the cube is solved.

With the cube in **Rubik** solved and the “FFRR” still visible in the “Current Macro” window, click on the $\boxed{\text{Apply Macro}}$ button. This will instantly apply your four moves and show you the result. If you wish, apply the same FFRR operation to your physical cube and compare the results. Click the same $\boxed{\text{Apply Macro}}$ button again and again until the cube returns to solved. It should be the same number of times as the order you calculated twice before. In fact, what **Rubik** is doing is just that—it starts with a solved cube, applies the move combination in the window time after time, and after each application, it checks to see if the cube is solved. When the cube has returned to the solved configuration, the order is simply the number of times that it took.

By the way, just so you don’t get mixed up, it might be a good idea to return your cube to solved now with rrff.

It’s not too much fun just to use the $\boxed{\text{Apply Macro}}$ button in **Rubik**—the cube just jumps to the result and you can’t see how it got there. Reset the cube again, and make sure that the same FFRR is in the “Current Macro” window, click in that “Current Macro” window with the mouse just before the first “F” and press the **return** key repeatedly on your keyboard. **Rubik** then twists the cube faces as you watch.

Note: if the cube faces turn too quickly or too slowly, see the **Rubik** documentation to learn how to set the turning speed to a reasonable value for your computer.

The **return** key is the scenic route and the $\boxed{\text{Apply Macro}}$ button is the superhighway.

Why is it the case that *any* cube operation, if repeated enough times, will eventually return to where it started?

Each time an operation is repeated, the facelets are rearranged. Since there are only a finite (although *very* large) number of possible rearrangements, we know that if we repeat the operation at least that number of times, we are guaranteed eventually to repeat one of the arrangements. This does not prove yet that the cube will return to the initial configuration, but at least it will repeat some arrangement.

Let’s call the operation P, where P stands for any combination of cube face twists. If we apply P repeatedly, eventually it will arrive at a repeat arrangement of the cubie facelets. Suppose that this first happens after m times and that this arrangement is the same as one that occurred at and earlier step k, where $k < m$. Thus $\mathsf{P}^k = \mathsf{P}^m$, and m is the smallest such number. Thus, unless $k = 0$, $\mathsf{P}^{k-1} \neq \mathsf{P}^{m-1}$. If $k = 0$, we are done, since $\mathsf{P}^0 = \mathsf{1}$, so suppose that $k > 0$.

Since $\mathsf{P}^k = \mathsf{P}^m$, this means that if we apply P either k times or m times to the same initial cube, we arrive at the same final cube arrangement. If we apply P^{-1} to that arrangement, the result will be the same, no matter whether it was arrived at after m or k steps. (Since applying the same operation to the same arrangement will yield the same result.)

But applying P^{-1} at the end of each exactly undoes the final application of P that was done.

If you apply P m times and then undo P once, that's the same as just applying it $m - 1$ times and similarly for k. Thus $\mathsf{P}^k\mathsf{P}^{-1} = \mathsf{P}^{k-1}$ and $\mathsf{P}^m\mathsf{P}^{-1} = \mathsf{P}^{m-1}$. Therefore $\mathsf{P}^{k-1} = \mathsf{P}^{m-1}$, contradicting the assumption that m was the smallest value where the rearrangement repeats. Thus k must be equal to 0, so $\mathsf{P}^m = \mathsf{1}$.

This is very interesting for a couple of reasons. From a purely artistic viewpoint, if you take a solved cube and repeatedly apply the same set of operations, it will eventually return to solved again and again. The **Rubik** program has a Demo button that causes the cube to spin through a random set of states but it keeps repeating that pattern, so if you watch long enough, it is guaranteed to return to solved, not once, but over and over again[1].

If you have your own favorite set of moves that goes through a bunch of pretty patterns, you can force **Rubik** to use that as its demo pattern. Simply type it into the "Current Macro" box before you press the Demo button. For example, try typing RlUdRb as your macro and then run the demo. If there's anything in that box, **Rubik** uses it for the demo pattern; if the box is empty, **Rubik** invents its own pattern.

But the fact that any pattern eventually returns to solved actually provides a brute-force mechanism that you could use to solve the cube, although your solution would be quite lengthy.

The usual method to solve a cube is to find combinations of moves, which, when applied as a unit (which is what we'll call a "macro"), do very specific things to the cube. For example, if you found a move that would flip two edge cubies in place, if the cube you were trying to solve had two edge cubies in that orientation, you could apply the macro and bring your cube one step closer to solution. In fact, when **Rubik** first comes up there is a set of such useful macros loaded into the "Defined Macros" area. See the user's guide to learn exactly how to use these.

The question is, how do you discover these macros that do very small, specific things and leave most of the cube unaltered? It turns out (and we shall see why later) that if you have a macro with a certain order and you apply it for half or a third of that number of steps, the result is often a usable (although usually very long) macro.

As an example, consider the FFRR macro that we experimented with before. We found (in three different ways, hopefully) that the order of this macro is 6. The "large" divisors of 6 are 2 and 3, so you may find interesting macros by repeating the FFRR combination twice or three times.

To do this, reset the cube and type FFRR into the "Current Macro" window. If you press the Apply Macro button twice there's a sort of a nice pattern, but it moves far too many facelets to be useful for solving the cube. Press it a third time, however, and you'll see that the net result is that two pairs of edge cubies are exchanged and everything else remains exactly as it was before. Thus FFRRFFRRFFRR might be useful to you as you're solving a cube.

It's easy to look for such macros. Simply type in various (usually short) sets of moves and find the order of that operation. If that order is divisible by a small number like 2 or 3 or perhaps 5, try dividing the order by that number and applying the macro that number of times. There is a shortcut for doing this. Suppose you find a pattern that repeats every 90 moves (the macro

[1] In fact, the code in the demo routine selects by trial and error a combination of moves such that the total number of moves is tolerably small—less than 300—so if you run the demo mode, you are guaranteed that the positions will start to repeat in fewer than 300 moves.

FFLLBR, for example, has order 90). If you want to see what this does to the cube after 45 moves (for a total of $45 \times 6 = 270$ moves, which would be fairly painful to use), you can simply type the following into the “Current Macro” window: “45(FFLLBR)”. A number in front of a group of moves in parentheses tells **Rubik** to repeat the stuff inside that many times. These groupings can be nested, but this will not be too useful for finding cube-solving macros.

To see why this strategy might produce useful patterns, we will need to take a detour to learn something about the structure of permutations.

4 Permutations

A permutation is a rearrangement of a set of objects. Keep in mind that it is the *rearrangement* that’s the important part; usually not the objects themselves. In some sense, the permutation that exchanges items 1 and 2 in the set $\{1, 2, 3\}$ is the same as the permutation that exchanges A and B in the set $\{A, B, C\}$—the rearrangement is the same; it’s just that the names of the particular items are different in the two cases. In what follows, unless we are talking about Rubik’s cube, we’ll just consider the objects to be moved to be the numbers from 1 to N.

One way to think about permutations of N objects is to visualize a set of boxes numbered 1 to N, and a set of balls with the same numbers 1 to N, where each box contains a single ball. A permutation consists of taking the balls out of the boxes and putting them back, either in the same or different boxes, so that at the end, each box again contains exactly one ball.

A permutation can be described by a series of statements like the following:

> The ball originally in box 1 is moved to box A_1.
> The ball originally in box 2 is moved to box A_2.
> The ball originally in box 3 is moved to box A_3.
> …et cetera.

The A_1, A_2, A_3, and so on represent numbers from 1 to N.

If the situation before the rearrangement occurs has ball number i in box number i for every i, then if we simply list the contents of the boxes in order, we have a complete description of the permutation.

As a concrete example, if the objects are 1, 2, 3 and 4, we might use 1342 to represent the permutation that leaves the contents of box 1 fixed, moves the ball in box 2 to box 3, from box 3 to box 4 and from box 4 to box 1.

The description above works because there is a natural order of the objects 1, 2, 3 and 4 but there is no such natural order to the cubies or faces in Rubik’s cube—does a yellow face “naturally” come before a red face? Who knows?

This problem can be solved by listing a permutation as two rows where the item in the top row is represents each original box and the item directly below it is the box to which the contents of that original box were moved. Thus the example permutation of the four numbers above can be described equally well by any of the following:

$$\binom{1234}{1342} \text{ or } \binom{2134}{3142} \text{ or } \binom{4321}{2431} \text{ or } \binom{3412}{4213} \tag{1}$$

or in any of 20 other forms, as long as there's always a 1 under the 1, a 3 under the 2, et cetera.

5 Permutation Cycle Notation

The notation introduced in the previous section certainly works to describe any permutation, but there is a much better way that we will call "cycle notation". If we are looking at a particular permutation, we can begin at any box and see where the contents of that box are moved by the permutation. If that first ball doesn't remain fixed, it moves to a new box, so the ball in that new box is moved to yet another box, and so on. Eventually, a ball has to move back to the original box since there are only a finite number of boxes. This forms a cycle where each ball moves to the next position in the cycle and the moves eventually "cycle around" to the original box.

These cycles can have any length from 1 up to N, the total number of boxes. In the previous example shown in equation 1, item 1 forms a cycle of length 1 (since it doesn't move, or if you like, it moves to itself). The other three form a cycle: 2 moves to 3, 3 moves to 4 and 4 moves back to 2. The cycle notation for that permutation is this:

$$(1)(2\ 3\ 4).$$

To interpret cycle notation, the set of items between each pair of parentheses form a cycle, with each moving to the box of the one that follows it. Finally, the last one in the list moves back to the box represented by the first one in the list. These cycles will be disjoint in the sense that each item will appear in only one of them. If an item appeared in two different cycles, then it would appear to follow two different paths.

Notice also that the cycle notation is not unique although it can be made to be. All the permutations in the list below are equivalent:

$$(1)(2\ 3\ 4) \quad (1)(3\ 4\ 2) \quad (1)(4\ 2\ 3) \quad (2\ 3\ 4)(1) \quad (3\ 4\ 2)(1) \quad (4\ 2\ 3)(1)$$

Since they are independent, we can list the cycles in any order, and since we can begin with any element in the cycle and follow it around, a cycle of n objects can appear in any of n forms.

5.1 Canonical Cycle Notation

This makes it a bit difficult to determine at a glance whether two descriptions of a permutation in cycle notation are equivalent, but if there is some sort of "natural" ordering to the objects then it is possible to form a canonical cycle notation:

1. Find the smallest item in the list and begin a cycle with it.

2. Complete this first cycle by following the movement of the objects by the permutation and close the cycle.
3. If you have finished listing all of the objects in the permutation, you are done; otherwise, return to step 1.

The canonical form of the cycle above is $(1)(2\ 3\ 4)$.

Let's now look at a few more complex permutations and see what their cycle notations look like.

The permutation $(1\ 3)(2\ 4)(5)$ exchanges the contents of boxes 1 and 3 and also exchanges the contents of boxes 2 and 4 and leaves the contents of box 5 unchanged.

The permutation $(1\ 2\ 3\ 4\ 5)(9\ 8\ 7)$ cycles 1 to 2, 2 to 3, 3 to 4, 4 to 5 and 5 back to 1. In addition, it cycles 9 to 8, 8 to 7 and 7 back to 9.

Often, if the set of objects being permuted is obvious, the objects that do not move are not listed. Thus $(1)(2\ 3\ 4)$ might be listed simply as $(2\ 3\ 4)$. With this convention, however, there's no reasonable way to list the identity permutation that moves nothing, so it is often listed as (1), where only one example of a non-moving object is listed, or even as 1 to indicate that it is an identity transformation.

If you were listing the primitive cube operations in this cycle notation, the convention of leaving out 1-cycles would be a big advantage. Of the 54 facelets on a cube, a single face twist only moves 21 of them, which obviates listing 33 of the 1-cycles.

5.2 The Cycle Structure of a Permutation

A very important feature of a permutation is captured when it is listed in cycle notation, and that is its cycle structure. For example, the cycle structure of

$$(1)(2)(3\ 4\ 5)(6\ 7\ 8)(9\ 10\ 11\ 12)$$

has two 1-cycles, two 3-cycles, and one 4-cycle. To see why this is important, let's begin with a few simple examples.

Consider $(1\ 2\ 3)$. If this operation is applied three times, it is obvious that the result is the identity permutation. Each time it is applied, each element advances to the next box in the cycle, but the cycle is three boxes long, so after three steps, each object will return to where it started. In fact, if P is a permutation whose structure consists of a single n cycle: $(i_1\ i_2\ i_3\ \cdots\ i_n)$ then $P^n = 1$.

Also obvious, but worth stating, is that if you apply a permutation that consists of a single cycle of length n repeatedly, it will return to the identity after *every* n applications. If P consists of a single 5-cycle, then $P^5 = P^{10} = P^{15} = P^{20} = \cdots = 1$.

Next, let's consider the permutation $P = (1\ 2\ 3)(4\ 5\ 6\ 7)$ that consists of both a 3-cycle and a 4-cycle. Since the two cycles have no elements in common, if we apply P repeatedly and don't pay any attention to the elements in the 4-cycle, we see the elements in the 3-cycle returning to their initial locations every three applications. Similarly, if we ignore the elements in the 3-cycle and pay attention only to those in the 4-cycle, then every 4 applications of P returns those four elements to their starting places.

In other words, the elements in the 3-cycle return to their original locations for P^3, P^6, P^9, P^{12}, P^{15}, and so on. Similarly, the elements in the 4-cycle return to their original locations for P^4, P^8, P^{12}, P^{16}, and so on.

Notice that P^{12} appears in both lists, and that this is the first exponent of P that is in both lists. This means that after 12 applications of P, the elements in both the 3-cycle and in the 4-cycle are returned to their starting locations, and furthermore, this is the first time that it happens. Thus $P^{12} = 1$, and since it's the first time this happens, the order of P is 12.

The number 12 is the least common multiple of 3 and 4, usually written as

$$\mathrm{LCM}(3, 4) = 12.$$

In other words, 12 is the smallest number larger than 0 that is a multiple of both 3 and 4. It should be obvious from the discussion above that if a permutation consists of two cycles of lengths m and n, then then order of that permutation is simply $\mathrm{LCM}(m, n)$.

The concept of a least common multiple can be extended easily to any number of inputs. We have: $\mathrm{LCM}(4, 5, 8, 7) = 280$—$280$ is the smallest number that is a multiple of 4, 5, 8 and 7. If a permutation consists of a 4-cycle, a 5-cycle, an 8-cycle and a 7-cycle, then the order of that permutation would be 280.

5.3 Applications of Cycle Structure to the Cube

Let's consider a permutation that looks like this: $P = (1\ 2)(3\ 4\ 5\ 6\ 7)$ that consists of a 2-cycle and a 5-cycle, so its order is $\mathrm{LCM}(2, 5) = 10$. What happens if we repeat P five times? In other words, what does P^5 look like? The 5-cycle will disappear, since after 5 applications, every element in it has cycled back to its starting point. The 2-cycle will have been applied an odd number of times, so it will remain a 2-cycle. Thus $P^5 = (1\ 2)$.

Thus, although the permutation P by itself moves 7 objects, the permutation P^5 moves only two objects. If the objects 1, 2, . . . , 7 in this example were really cubies in Rubik's cube, then if the operation P were repeated 5 times, the net result would be an operation that moved exactly 2 cubies and left all the others where they were.

Toward the end of Section 3.1 we saw an example of this: The operation FFRR moves 13 cubies, but (FFRR)^3 moves only 4—it exchanges two pairs of edge cubies.

Using the names for the individual cubies described in the documentation for the **Rubik** program, here is what the permutation FFRR does:

(DF UF)(DR UR)(BR FR FL)(DBR UFR DFL)(ULF URB DRF)

where we use "DF" to indicate the "down-front" edge cubie, "DBR" to represent the "down-back-right" cubie, et cetera. Obviously, since there are 13 cubies in the permutation cycle listing, 13 of the cubies are moved by FFRR. But nine of those 13 appear in 3-cycles, so the permutation

$(\mathsf{FFRR})^3$ leaves those nine cubies fixed, moving only the two pairs of edge cubies that we noticed earlier.

The **Rubik** program has a command “Display Permutation” in the “File” pull-down menu that will display the permutation that is required to get to the current cube coloring from the solved cube. Although the notation above appears to describe the permutation, there are a couple of problems with it:

1. If a cubie is left in its same position but is rotated (a corner cubie) or flipped (an edge cubie), then there is no way to indicate this.
2. Even if the cubies move in a cycle to different positions on the cube, there is again no way to indicate how they are flipped or rotated in their new positions relative to how they were before.

The easiest way to indicate the details of a permutation exactly is to list where every facelet of every cubie moves. Assuming that the center cubies stay in place, there are 48 of these facelets that can move so such a complete description is a lot longer, and doesn’t make it quite so obvious which cubies move to which locations.

So in spite of its drawbacks, the first form of the notation is usually the most useful. It can be improved slightly so that it will indicate cubies that are flipped or rotated in place as follows: (UF) means that the up-front edge cubie stays in place, but is flipped. (URF) means that the up-right-front corner cubie is twisted in place where the up facelet moves to the right facelet, the right to the front, and the front back to the up facelet.

When you issue the “Display Permutation” command you will be presented with both the most useful and most accurate descriptions of the permutations. The notation for indicating the movement of cubie facelets requires that each corner cube be assigned the names for its three facelets and each edge cubie needs two. The (URF) corner cubie has the following three facelets: (Urf), (Ruf), and (Fur). The three letters indicate which corner it is, and the letter in upper case is the particular cubie facelet. Similarly, the two facelets of the cubie (UF) are (Uf) and (Fu). The description of the movement that flips the UF and UL cubies in place is this: (Lu Ul)(Fu Uf).

To save you the trouble of counting the number of terms in each cycle, the cycle notation is listed below each permutation. A permutation having the following notation:

4(3) 2(5) 1(6) 1(8) 1(12)

means that the particular permutation consists of four 3-cycles, two 5-cycles, and one each of a 6-cycle, an 8-cycle and a 12-cycle.

In the first listing **Rubik** may also list a certain number of 1-cycles, but these simply represent the number of cubies that say in place and are either flipped or rotated. Look at the detailed permutation description to see what they are. Cubies that stay in place and are not moved are not listed as 1-cycles. Similarly, the 1-cycles in the face-permutation listing are not included.

Here is an example where the permutation cycle form can be used to find a macro that would be truly useful for solving the cube, although it contains far too many steps. On the other hand, if you didn't know any better techniques, this one would work. The example also illustrates one of the shortcomings of the cubie-based cycle notation. Although you apply a 9-cycle nine times, it does not return completely to solved, since those movements have a net effect of flipping some edge cubies. If you look at the cubie-facelet permutation you will see that one of the cycles in fact has length 18.

Imagine that you've experimented with a number of short move sequences and you stumble across this one: FUllR. You find that the order of this permutation is 36, but when you look at the cycle notation, you obtain this:

(UR UF)(UL UB BR DR FR DF BL FL DL)
(RFU)(BRU)(DRF UBL DBR)(DLB ULF)(DFL)

Its cycle structure contains cycles of lengths 9, 3, and 2. At first it looks like applying it 9 times might be useful since that would only leave a pair of 2-cycles, but when you try this, you obtain:

(UR UF)(UB)(UL)(DF)(DR)(DL)(FR)(FL)(BR)(BL)(DLB ULF)

In fact, the long cycles also flip cubies when they operate, so far too much is done by this operation. However, we noticed that the order of the macro was 36, not 18, and thus if we do 9 more applications, it will undo the flips and it must leave something changed afterwards. When we do this, the cycle structure is simply:

(UF)(UR)

which flips two cubies in place. The unfortunate thing is that 18 applications of a 5-step macro or 90 total twists are required to do this.

6 What Is a Group?

A group is an abstract mathematical object that can be defined in terms of a few simple axioms and about which theorems can be proved. The set of permutations of Rubik's cube provide an example of a group, but unfortunately, of a large and fairly complex group.

We will be able to use some properties of group theory to manipulate the cube, but, as before, if we want to learn something about groups, it is a good idea to begin looking at simple ones with only a few members; the group $\mathcal{R}$ corresponding to Rubik's cube has $8!12!2^{10}3^7 = 43252003274489856000$ members, one corresponding to each position reachable from a solved cube. It's probably easier to begin by looking at groups with 2 or 4 or 6 members.

6.1 Formal Definition

A group $\mathcal{G}$ consists of a set of objects and a binary operation $*$ on those objects satisfying the following four conditions:

1. The operation $*$ is closed. In other words, if g and h are any two elements of the group $\mathcal{G}$ then the object $g * h$ is also in $\mathcal{G}$.
2. The operation $*$ is associative. In other words, if f, g and h are any three elements of $\mathcal{G}$, then $(f * g) * h = f * (g * h)$
3. There is an identity element e in $\mathcal{G}$. In other words, there exists an $e \in \mathcal{G}$ such that for every element $g \in \mathcal{G}$, $e * g = g * e = g$.
4. Every element in G has an inverse relative to the operation $*$. In other words, for every $g \in \mathcal{G}$, there exists an element $g^{-1} \in \mathcal{G}$ such that $g * g^{-1} = g^{-1} * g = e$.

For those who desire the absolute minimum in conditions, see the footnote[2].

Notice that one of the properties that you are used to in most systems is not necessarily present in a group: commutativity. In other words, there may exist elements g and h of $\mathcal{G}$ such that $g * h \neq h * g$. Notice also that the definition about says nothing about a group being finite, although in this paper we will consider mostly finite groups, although in the case of the $\mathcal{R}$, very large finite groups.

Since there is only one operation $*$ we often omit it and write gh in place of $g * h$. Similarly, we can define $g^2 = gg = g * g$, $g^3 = ggg = g * g * g$ and so on. $g^0 = e$, and $g^{-n} = (g^{-1})^n$. Because of associativity, these are all well-defined.

6.2 Examples of Groups

You are already familiar with a few groups, but most of the best-known groups are infinite: the integers under addition, the rational numbers under addition, the rational numbers except for 0 under multiplication, the real numbers or complex numbers under addition, the real or complex numbers except for 0 under multiplication. All of these groups are infinite and commutative. (Commutative means that $a*b = b*a$ for every a and b in the group.) A group that is commutative is called an abelian group.

The natural numbers ($= \{0, 1, 2, 3, \ldots\}$) under addition do not form a group—there is an identity (0), but there are no inverses for any positive numbers. We can't include zero in the rational, real, or complex numbers under multiplication since it has no inverse.

The so-called trivial group consists of one element, 1, and satisfies $1 * 1 = 1$ is the simplest group. Since a group has to contain the identity element, the trivial group is the smallest group possible.

If you know about modular arithmetic, then if the operation is addition modulo n, the n elements 0, 1, …, $n - 1$ form a group under that operation. This is a finite commutative group. If p is prime, then multiplication modulo p forms a group containing $p - 1$ elements: 1, 2, …, $p - 1$. If p is not a prime then the operation does not form a group. For example, if $p = 6$ there

[2] In fact, there is a slightly simpler and equivalent definition of a group. Only a right identity and a right inverse are required (or a left identity and a left inverse). In other words, if there is an e such that $g * e = g$ for all $g \in \mathcal{G}$ and for every $g \in \mathcal{G}$ there exists a g^{-1} such that $g * g^{-1} = e$ then you can show that $e * g = g$ and that $g^{-1} * g = e$. This can be done by evaluating the expression $g^{-1} * g * g^{-1} * (g^{-1})^{-1}$ in two different ways using the associative property.

is no inverse for 2: $2 * 1 = 2$, $2 * 2 = 4$, $2 * 3 = 0$, $2 * 4 = 2$ and $2 * 5 = 4$. (Remember that the "$*$" represents multiplication modulo 6.) When two numbers, neither of which is zero, multiply to yield zero, then the system is said to have zero divisors. When a modular system under multiplication has no zero divisors it forms a group.

In the group based on addition modulo n, if you begin with the element 1, you can get to any element in the group by successive additions of that element. In the group of order 5, you have: $1 = 1$, $2 = 1 + 1$, $3 = 1 + 1 + 1$, $4 = 1 + 1 + 1 + 1$ and $0 = 1 + 1 + 1 + 1 + 1$. The same idea holds for any n. In this case we say that the group is generated by a single element (1 in this case), and such groups are called cyclic groups, since successive additions simply cycle through all the group elements. For this same group corresponding to $n = 5$, the element 3 is also a generator: $1 = 3 + 3$, $2 = 3 + 3 + 3 + 3$, $3 = 3$, $4 = 3 + 3 + 3$ and $5 = 3 + 3 + 3 + 3 + 3$. Does that group have any other generators?

For any particular geometric object, the symmetry operations form a group. A symmetry operation is a movement after which the object looks the same. For example, there are 4 symmetry operations on an ellipse whose width and height are different:

1: Leave it unchanged
a: Rotate it $180°$ about its center
b: Reflect it across its short axis
c: Reflect it across its long axis

The group operation consists of making the first movement followed by making the second movement. Clearly 1 is the identity, and each of the operations is its own inverse. We can write down the group operation $*$ on any pair of elements in the following table:

$*$	1	a	b	c
1	1	a	b	c
a	a	1	c	b
b	b	c	1	a
c	c	b	a	1

The group of symmetries of an equilateral triangle consists of six elements. You can leave it unchanged, rotate it by $120°$ or $240°$, and you can reflect it across any of the lines through the center and a vertex.

In the same way, the group of symmetries of a square consists of eight elements: the four rotations (including a rotation of $0°$ which is the identity) and four reflections through lines passing through the center and either perpendicular to the edges or the diagonals. In general, a regular n-gon has a group of $2n$ symmetries which is usually called the dihedral group.

A circle has an infinite number of symmetries. It can be rotated about its center by any angle θ such that $0 \leq \theta < 2\pi$ or it can be reflected across any line passing through its center.

6.3 Permutation Groups

The most important example (since we're supposed to be fixated on Rubik's cube as we read this) is that certain sets of permutations also form groups. Since a permutation is just a rearrangement of objects, the group operation is simply the concatenation of two such rearrangements. In other words, if g is one rearrangement and h is another, then the rearrangement that results from taking the set of objects and applying g to it, and then applying h to the rearranged objects is what is meant by $g * h$.

To avoid a possible misunderstanding, when we speak about the Rubik's cube group, the group members are move sequences and the single operation is the act of doing one sequence followed by another. At first it's easy to get confused if you think of rotating the front face as a group operation. The term "move sequence" above is not exactly right either—move sequences that have the same final result are considered to be the same. For an easy example, F and F^5 are the same group element.

The Rubik's cube group is simply the set of all possible permutations of the facelets achievable with twists of the cube faces. To combine two of these permutations, we simply apply one after the other. This, of course, is a huge group. Since this group is so prominent in this paper, we'll give it a special name: $\mathcal{R}$.

In any permutation group the identity permutation that leaves all the objects in place will, of course, be the group identity. The inverse of a permutation is the permutation that exactly undoes it. To multiply two permutations together, just pick each element from the set of objects being permuted and trace it through. For example, if the set of objects that are to be permuted consists of the six objects $\{1, 2, 3, 4, 5, 6\}$ and we wish to multiply together $(1\ 2\ 4)(3\ 6)$ and $(5\ 1\ 2)(4\ 3)$ we can begin by seeing what happens to the object in box 1 under the influence of the two operations. The first one moves it to box 2 and the second moves the object in box 2 to box 5. Thus, the combination moves the object in box 1 to box 5. Therefore, we can begin to write out the product as follows:

$$(1\ 2\ 4)(3\ 6) * (5\ 1\ 2)(4\ 3) = (1\ 5\ \ldots$$

We write "..." at the end since we don't know where the object in box 5 goes yet. Let's trace 5 through the two permutations. The first does not move 5 and the second moves 5 to 1, so $(1\ 5)$ is a complete cycle in the product.

Here's what we have, so far:

$$(1\ 2\ 4)(3\ 6) * (5\ 1\ 2)(4\ 3) = (1\ 5)\ldots$$

We still need to determine the fates of the other objects. So far, we haven't looked at 2, so let's begin with that. The first permutation takes it to 4 and the second takes 4 to 3 so we've got this:

$$(1\ 2\ 4)(3\ 6) * (5\ 1\ 2)(4\ 3) = (1\ 5)(2\ 3\ \ldots$$

Doing the same thing again and again, we find that the pair of permutations takes 3 to 6, that it takes 6 to 4, and finally, it takes 4 back to 2. Thus the final product of the two permutations is given by:

$$(1\ 2\ 4)(3\ 6) * (5\ 1\ 2)(4\ 3) = (1\ 5)(2\ 3\ 6\ 4).$$

From now on we'll omit the "$*$" operator and simply place the permutations to be multiplied next to each other. As an exercise, verify the following product of permutations of the 9 objects $\{1, 2, \ldots, 9\}$:

$$(1\ 2\ 3)(4\ 5)(6\ 7\ 8\ 9)\ (2\ 5\ 6)(4\ 1)(3\ 7) = (1\ 5)(2\ 7\ 8\ 9)(3\ 4\ 6).$$

As we noticed when we looked at permutations of the facelets of Rubik's cube, the order makes a difference: $(1\ 2)(1\ 3) \neq (1\ 3)(1\ 2)$ since $(1\ 2)(1\ 3) = (1\ 2\ 3)$ and $(1\ 3)(1\ 2) = (1\ 3\ 2)$.

Let's look in detail at a particular group—the group of all permutations of the three objects $\{1, 2, 3\}$. We know that there are $n!$ ways to rearrange n items since we can chose the final position of the first in n ways, leaving $n-1$ ways to chose the final position of the second, $n-2$ for the third, and so on. The product, $n \cdot (n-1) \cdot (n-2) \cdots 3 \cdot 2 \cdot 1 = n!$ is thus the total number of permutations. For three items that means there are $3! = 6$ permutations:

$$(1),\ (1\ 2),\ (1\ 3),\ (2\ 3),\ (1\ 2\ 3) \text{ and } (1\ 3\ 2).$$

Table 1 is a "multiplication table" for these six elements. Since, as we noted above, the multiplication is not necessarily commutative, the table is to be interpreted such that the first permutation in a product is chosen from the row on the top and the second from the column on the left. At the intersection of the row and column determined by these choices is the product of the permutations. For example, to multiply $(1\ 2)$ by $(1\ 3)$ choose the item in the second column and third row: $(1\ 2\ 3)$.

	(1)	$(1\ 2)$	$(1\ 3)$	$(2\ 3)$	$(1\ 2\ 3)$	$(1\ 3\ 2)$
(1)	(1)	$(1\ 2)$	$(1\ 3)$	$(2\ 3)$	$(1\ 2\ 3)$	$(1\ 3\ 2)$
$(1\ 2)$	$(1\ 2)$	(1)	$(1\ 3\ 2)$	$(1\ 2\ 3)$	$(2\ 3)$	$(1\ 3)$
$(1\ 3)$	$(1\ 3)$	$(1\ 2\ 3)$	(1)	$(1\ 3\ 2)$	$(1\ 2)$	$(2\ 3)$
$(2\ 3)$	$(2\ 3)$	$(1\ 3\ 2)$	$(1\ 2\ 3)$	(1)	$(1\ 3)$	$(1\ 2)$
$(1\ 2\ 3)$	$(1\ 2\ 3)$	$(1\ 3)$	$(2\ 3)$	$(1\ 2)$	$(1\ 3\ 2)$	(1)
$(1\ 3\ 2)$	$(1\ 3\ 2)$	$(2\ 3)$	$(1\ 2)$	$(1\ 3)$	(1)	$(1\ 2\ 3)$

Table 1: Multiplication of permutations of 3 objects

If we make a similar table of the symmetries of an equilateral triangle $\triangle ABC$ (with A, B and C listed counter-clockwise) as described above whose elements are 1, rotate $120^\circ = R1$, rotate $240^\circ = R2$, flip across axis A, B or C (FA, FB, FC), then you would obtain table 2.

If you look carefully at tables 1 and 2, you can see that they are really the same—the only difference is the names of the permutations. If you substitute 1 for (1), FA for $(1\ 2)$, FB for $(1\ 3)$, FC for $(2\ 3)$, $R1$ for $(1\ 2\ 3)$ and $R2$ for $(1\ 3\ 2)$, the two tables are identical, so in a sense, the two groups are identical and we call them isomorphic.

	1	FA	FB	FC	$R1$	$R2$
1	1	FA	FB	FC	$R1$	$R2$
FA	FA	1	$R2$	$R1$	FC	FB
FB	FB	$R1$	1	$R2$	FA	FC
FC	FC	$R2$	$R1$	1	FB	FA
$R1$	$R1$	FB	FC	FA	$R2$	1
$R2$	$R2$	FC	FA	FB	1	$R1$

Table 2: Multiplication of symmetries of an equilateral triangle

In fact, it is easy to see why this is the case. The symmetries of $\triangle ABC$ just move the letters labeling the vertices around to new locations and the six symmetries of the triangle can arrange them in any possible way, so in a sense, the triangle symmetries rearrange A, B and C and the permutation group rearranges the objects 1, 2 and 3.

This group that contains all the permutations of three objects is called the symmetric group on three objects. In general, the group consisting of all the permutations on n objects is called the symmetric group on n objects. Since there are n! permutations of n objects, that is the size of the symmetric group.

Thus when you read in your group theory text that there are exactly two groups of order 6, what this means is that every group, with an appropriate relabeling of the members of the group, will be like one of those two groups. When two groups have this relationship, we say that they are isomorphic. (The groups in tables 1 and 2 are the same; the other group of order six is the one corresponding to addition modulo 6, described in Section 6.2.)

A permutation group does not have to include all possible permutations of the objects. If we consider the $\mathcal{R}$ as a permutation group, there is obviously no permutation that moves an edge cubie to a corner cubie and vice-versa. The group consisting of the complete set of permutations of three objects shown in table 1 contains various proper subsets that also form groups:

$$\{1\}, \{1, (1\,2)\}, \{1, (1\,3)\}, \{1, (2\,3)\}, \text{ and } \{1, (1\,2\,3), (1\,3\,2)\}. \qquad (2)$$

These subsets of groups that are themselves groups under the same operation are called subgroups. The group $\mathcal{R}$ is a subgroup of the group of all permutations of 48 items.

6.4 Properties of Groups

This paper is not meant to be a complete course in group theory, so we'll list below a few of the important definitions and some properties satisfied by all groups, the proofs of which can be found in any elementary introduction to group theory or abstract algebra.

1. The identity is unique and every element of $\mathcal{G}$ has a unique inverse.
2. The order of an element $g \in \mathcal{G}$ is the smallest positive integer n such that $g^n = e$. In a finite group every element has a finite order.

3. The order of a group is the number of elements in it. if $g \in \mathcal{G}$ then the order of g divides evenly the order of $\mathcal{G}$.

4. We say that $\mathcal{H}$ is a subgroup of a group $\mathcal{G}$ if $\mathcal{H} \subset \mathcal{G}$ and $\mathcal{H}$ itself is a group under the same binary operation $*$ that's used in $\mathcal{G}$. If $\mathcal{H}$ is a subgroup of $\mathcal{G}$ then the order of $\mathcal{H}$ divides evenly into the order of $\mathcal{G}$.

5. If $\mathcal{H}$ and $\mathcal{K}$ are both subgroups of the same group $\mathcal{G}$, then $\mathcal{H} \cap \mathcal{K}$ is also a subgroup of $\mathcal{G}$.

Using as an example the symmetric group on three objects displayed in table 1, the order of $(1\ 2)$ is 2, the order of $(1\ 2\ 3)$ is 3, and both 2 and 3 divide 6, the order of the group. The proper subgroups of the symmetric group listed in equation 2 have orders 1, 2, and 3—again, all divisors of 6, as they should be.

Any pair of subgroups in that list only have the identity element in common, so clearly the intersection of any two of them is also a group, although it is the trivial group.

If we look at the symmetric group $\mathcal{G}$ on 4 objects (the group of order $4! = 24$ that contains all the permutations of 4 objects), let $\mathcal{H}$ be the subgroup of $\mathcal{G}$ that consists of all permutations that leave the element 1 fixed (but with no further restrictions), and let $\mathcal{K}$ be the set of permutations that leave 2 fixed.

Then we have:

$$\begin{aligned}
\mathcal{H} &= \{(1), (2\,3), (2\,4), (3\,4), (2\,3\,4), (2\,4\,3)\} \\
\mathcal{K} &= \{(1), (1\,3), (1\,4), (3\,4), (1\,3\,4), (1\,4\,3)\} \\
\mathcal{H} \cap \mathcal{K} &= \{(1), (3\,4)\},
\end{aligned}$$

illustrating that the intersection of two subgroups is also a subgroup (and in this case, the intersection is the set of all permutations that leave both 1 and 2 fixed).

For the symmetric permutation groups, it is easy to see why the order of an element has to divide the order of the group. As we saw in Section 5.1, we can write any particular permutation down as a set of cycles, and the order of that permutation is simply the least common multiple of the cycle lengths. Since there are n elements that are moved by the permutations, the longest cycle can have length at most n, so all the cycle lengths are thus n or less. But the order of the group is $n!$, so clearly the LCM of a set of numbers less than n will divide $n!$.

7 Simple Subgroups of the Rubik Group

In its total glory, a jumbled Rubik's cube is difficult to unjumble, especially when you are a beginner. A common method to learn about complex situations is to look first at simpler cases and learn as much as you can about them before tackling the harder problem.

One way to simplify Rubik's cube is to consider only a subset of moves as being allowable and to learn to solve cubes that were jumbled with only those moves. If you do this, you are effectively reducing the number of allowable permutations, but you will still be studying a subgroup of the full Rubik group.

Let's consider a few subgroups. You may wish to investigate these yourself. The **Rubik** program contains a "macro gizmo" that may make this easier. If you would like to investigate the positions achievable using a limited set of moves, define each of those moves as a macro and put all of them in the macro gizmo. Then make moves from an initialized cube using only macro gizmo entries. In fact, if you place the macro gizmo on top of the control panel of **Rubik**, you will not press any other buttons by accident. If you restrict your moves to any of these subgroups, the cube will be easier to solve.

The list below is a tiny subset of the total number of subsets of the whole group, but these are "practical" examples in that you can experiment with a real cube making only the moves in the indicated subgroups. In a later section, we will examine in more detail more general (but less practical) subgroups of $\mathcal{R}$

1. **Single face subgroup.** In this subgroup you are only allowed to move a single face. This group is not very interesting, since there are only 4 achievable positions including "solved", but it is a proper subgroup of the whole group.

2. **Two opposite faces subgroup.** This is also a fairly trivial group since twists of two opposite faces are independent. Still, it has 16 elements and is an example of what is known as a direct product group. **Beware:** if you are allowed to turn two adjacent faces, the subgroup is enormous: 73483200 members.

3. **F-R half-turn subgroup.** In this group, you are allowed to move either the front face or the right face by half-turns. This subgroup is of order 12 and we have already done a bit of analysis of this situation in Section 3.1.

4. **The slice subgroup.** In this group, you can only move the center slices. The subgroup can be further restricted by requiring that one, two, or three of those slices must make half-turns only. The full slice group contains 768 members. If one of the slices must be a half-turn, there are 192 members. If two are half-turns, there are 32 group members, and if all three moves must be half-turns, there are only 8 members.

8 How Many Cube Positions Can Be Reached?

> Ideal Toy Company stated on the package of the original Rubik cube that there were more than three billion possible states the cube could attain. It's analogous to MacDonald's proudly announcing that they've sold more than 120 hamburgers.
>
> —J. A. Paulos, *Innumeracy*

In Section 6 we said that the total number of reachable positions from a solved cube is the following huge number: $8! \cdot 12! \cdot 2^{10} \cdot 3^7 = 43252003274489856000 = 2^{27} \cdot 3^{14} \cdot 5^3 \cdot 7^2 \cdot 11$. How was this calculated? That's what we'll investigate in this section, but we'll need to learn to use some mathematical tools to do so. We can also investigate, with these same tools, the orders of some of the subgroups of the full cube group $\mathcal{R}$.

8.1 Even and Odd Permutations

In this section we will show that all permutations can be divided into two groups—those with even and odd parity. Just as is the case of addition of whole numbers, combining two permutations of even parity or two of odd parity results in a permutation of even parity. If only one of the two has odd parity, the result is odd.

Notice the following:

$$\begin{aligned}
(1\,2) &= (1\,2)\\
(1\,2)(1\,3) &= (1\,2\,3)\\
(1\,2)(1\,3)(1\,4) &= (1\,2\,3\,4)\\
(1\,2)(1\,3)(1\,4)(1\,5) &= (1\,2\,3\,4\,5)\\
(1\,2)(1\,3)(1\,4)(1\,5)(1\,6) &= (1\,2\,3\,4\,5\,6)
\end{aligned}$$

and it is not hard to prove that the pattern continues. This shows that any n-cycle can be expressed as a product of 2-cycles. If n is even, there are an odd number of 2-cycles and vice-versa. Since every permutation can be expressed as a set of disjoint cycles, this means that every permutation can be expressed as a product of 2-cycles. For example:

$$(1\,4\,2)(3\,5\,6\,7)(9\,8) = (1\,4)(1\,2)(3\,5)(3\,6)(3\,7)(9\,8).$$

Obviously, there are an infinite number of ways to express a permutation as a product of 2-cycles:

$$(1\,2\,3) = (1\,2)(1\,3) = (1\,2)(1\,3)(1\,2)(1\,2) = (1\,2)(1\,3)(1\,2)(1\,2)(1\,2)(1\,2)\cdots$$

but it turns out that for any given permutation, the number of 2-cycles necessary is either always even or always odd. For this reason, we can say that a permutation is either even or odd, depending on whether the representation of that permutation requires an even or odd number of 2-cycles.

This is not too hard to prove. Suppose that we consider a permutation of the set $\{1, 2, \ldots, n\}$ that moves 1 to x_1, 2 to x_2, 3 to x_3 and so on. All the x_i are different, and are just the numbers from 1 to n in some order. Consider the product:

$$\prod_{1\leq j<i\leq n}(x_i - x_j) = (x_2 - x_1)(x_3 - x_1)\cdots(x_n - x_1)(x_3 - x_2)\cdots(x_n - x_{n-1}) \qquad (3)$$

If you have never seen the Π-product notation before, the Greek symbol Π (pi) in front indicates a collection of things to be multiplied. In the example above, it means to multiply together all possible terms of the form $(x_i - x_j)$ where $1 \leq i < j \leq n$. It heightis similar to the Σ notation for summation, if you have seen that before. If you find it easier to understand, the product notation above has the following alternate representation where both i and j step up one at a time:

$$\prod_{1\leq j<i\leq n}(x_i - x_j) = \prod_{i=1}^{n-1}\Bigg(\prod_{j=i+1}^{n}(x_i - x_j)\Bigg)$$

Since all the x_i are different, every term in the product is non-zero, so the product itself is also non-zero, but it may be positive or negative. If the product is negative, we will call the permutation odd and if the product is even, we'll call it even.

First, let's check to see that the definition seems to make sense, at least in a few simple cases.

The permutation that swaps 1 and 2, $(1\ 2)$ has $x_1 = 2$ and $x_2 = 1$, so the product has only a single term: $(x_2 - x_1) = (1 - 2) = -1$ which is negative, so a permutation with one cycle (one is odd) corresponds to a negative product.

Now consider $(1\ 3\ 2)$. This should be an even permutation since $(1\ 3\ 2) = (1\ 3)(1\ 2)$ and thus the corresponding product should be positive. We have $x_1 = 3$, $x_2 = 1$ and $x_3 = 2$, and the calculation below shows that indeed the product is positive:

$$(x_2 - x_1)(x_3 - x_1)(x_3 - x_2) = (1 - 3)(2 - 3)(2 - 1) = (-2)(-1)(1) = +2 > 0.$$

You can check a couple more if you like, but you'll discover that it always seems to work, but why?

The identity permutation should be even (it can be represented by zero 2-cycles, and 0 is even). For the identity, $x_i = i$ for all i, so if $i > j$, $x_i - x_j = i - j > 0$, so all the terms in the product are positive, making the product positive.

Now, if we multiply any permutation by a 2-cycle, this should change it from even to odd or vice-versa, so we'd like to see that multiplying on a 2-cycle will flip the sign of the product. The following technique will work for any 2-cycle, but let's just look at multiplication of some permutation by the 2-cycle $(1\ 2)$.

This 2-cycle exchanges 1 and 2, so in the product, every x_1 becomes an x_2 and vice-versa. Let's write the product in the following form:

$$\prod_{1 \le j < i \le n} (x_i - x_j) = \begin{array}{r} (x_2 - x_1)(x_3 - x_1)(x_4 - x_1)\cdots(x_n - x_1) \\ (x_3 - x_2)(x_4 - x_2)\cdots(x_n - x_2) \\ (x_4 - x_2)\cdots(x_n - x_3) \\ \ddots \qquad \vdots\quad \\ (x_n - x_{n-1}) \end{array}$$

If we exchange x_1 and x_2, the sign of $(x_2 - x_1)$ will flip, but consider the rest of the line. Each term in the remainder of the line will become exactly the same as the term directly below it, and the term directly below will become that term, so there will be no additional changes of sign in the rest of the terms of the product. (In other words, when you substitute x_2 for x_1 in $(x_3 - x_1)$ it becomes $(x_3 - x_2)$ but when you substitute x_1 for x_2 in the term below it, $(x_3 - x_2)$, you obtain $(x_3 - x_1)$ so both substitutions together leave the product unchanged.) Hence, only one term changes sign, so the product will flip from positive to negative or vice-versa, completing the proof.

It is clear that if you look at all possible permutations of a set of objects, exactly half of them will have even parity and the other half, odd. In fact, an important subgroup of the symmetric group on n objects (the group of all possible permutations), the subset that consists of just the even permutations forms a subgroup called the alternating group on n objects. (Obviously, the subset of the odd permutations does not form a subgroup since it is missing the identity.) The alternating groups on 5 or more objects are the first examples of so-called simple groups that you will encounter in any formal class on group theory.

8.2 Parity and the Cube

We know that every possible permutation of the cube can be achieved by some combination of single turns of one face, and it is also easy to see that every face turn has even parity with respect to the movements of the cubies. The cycle structure for a single clockwise quarter-turn of the front face is this:

$$\text{(FL UP FR DF)(ULF UFR DRF DFL)}$$

which clearly has even parity since each of the 4-cycles can be written as a product of three 2-cycles for six total 2-cycles making the parity even. This means that there is no combination of moves of the cube that will exchange a single pair of cubies because that would correspond to an odd permutation of the cubies.

A cycle of three cubies of the same kind is possible, or an exchange of two pairs, both edges, both corners, or one of each. If the goal of solving Rubik's cube were simply to get the corner cubies and edge cubies into their correct positions but not to worry about whether they were oriented correctly, then if you were to break the cube apart and reassemble it at random, on average half of your reassemblies would result in a solvable cube.

But usual solution does require that you get the orientations of the edge and corner cubies correct, and it turns out that there are additional restrictions on these orientations. Let's consider first the orientations of the edge cubies where we will see that an even number of them must be flipped, so they, too, satisfy a parity condition.

Imagine a cube in outer space held in space such that the center cubies stay fixed as the other cubies turn around them. If you imagine a set of three-dimensional coordinate axes whose origin is at the center of the cube and that such that each axis goes through the center of a pair of center cubies, then for each axis, there are four edge cubies whose outer edge is aligned with that axis. Each axis has a positive and a negative direction, and let us mark the outer edge of each cubie with an arrow that is aligned with the positive direction of the axis parallel to it in the solved configuration.

At any stage, you can look at the arrows on each edge cubie's outer edge to see if they are aligned with their current axis. We will show that any single turn of a face changes the orientation of exactly two of them (an even number of them), so it is impossible with any number of twists to flip exactly one cubie in place.

Figure 8.2 illustrates the results of a $90°$ counter-clockwise rotation: the arrow configuration on the left will be converted to the arrow configuration on the right. It is clear that exactly two of the arrow directions will be flipped. Thus every turn of a face will flip exactly two arrows, so at any stage, an even number of the edge cubies will be flipped since in the original configuration

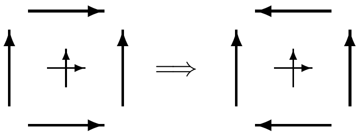


Figure 1: Preservation of Edge Parity

zero of them were flipped. (The crossing arrows in the middle of each face represent two of the fixed reference axes. The third axis points at you from out of the paper.)

The corner cubies satisfy a slightly different condition: the total rotation of all eight must be zero. Imagine the cube in a solved configuration, put a mark on the top face of all the top corner cubies and a mark on the bottom face of all the bottom corner cubies. After some number of twists of the cube faces, not all of the marks will be on the top and bottom of the cube.

If we look at any corner on a line of sight passing through the center of the cube, there are three possible orientations of each corner cubie: the mark can be on the top or bottom (in which case we will call its rotation $0°$) it can be rotated $120°$ clockwise (and call this rotation $120°$) or it can be rotated by $240°$ clockwise (called $240°$). The claim is that if you add all these rotation numbers for all the corner cubies, you will obtain a number that is a multiple of $360°$. In other words, the total rotation is a multiple of $360°$.

To see this, we can again look at what a single face turn does. If every face turn preserves this condition, then so will any combination of them. If you look at what happens with a single quarter-turn of a face, two of the faces are turned $120°$ clockwise from where they were before, and the other two are turned $120°$ counter-clockwise, which is the same as a $240°$ clockwise rotation. Thus a total rotation of $2 \times 120° + 2 \times 240° = 720°$ is applied to the four corner cubies, so we are done.

This means that if the cube were assembled randomly, only one third of the assemblies could be manipulated to put the corner cubes in a correct orientation. One third of the time you'd be off by a total of $120°$ and another third of the time you'd be off by $240°$.

OK, now we are finally in a position to count the total number of configurations that can be reached from a solved cube.

First, let's consider the number of configurations that could be constructed with no constraints. In other words, if you pop the cube apart with a screwdriver, how many ways can you put it together? There are 8 possible locations for each corner cubie, and if all possible arrangements were possible, there would be $8!$ rearrangements. Similarly, there are $12!$ rearrangements of the edge cubies. Each corner cubie could be in any of 3 rotations, so there are 3^8 ways of aligning the corner cubies and similarly there are 2^{12} flipping configurations of the edge cubies.

The grand total of configurations is thus: $8! \cdot 12! \cdot 3^8 \cdot 2^{12}$. But only $1/3$ of them will have the rotations of the corner cubies right, only $1/2$ of those will have the edge-flipping parity right, and only $1/2$ of those will have the correct cubie-rearrangement parity. Thus the total number of reachable configurations from a solved cube is:

$$(8! \cdot 12! \cdot 2^{12} \cdot 3^8)/(3 \cdot 2 \cdot 2) = 43252003274489856000.$$

A cube reassembled at random after breaking it apart would only have one chance in twelve of being solvable.

9 Change of Coordinates (Conjugation)

Suppose you have a block of metal and you need to drill a hole sideways through it. You have a drill press, but it only drills holes straight down. To drill your hole, you'd turn your block so that the side into which you need to drill is up, you'd drill the hole, and finally turn your piece back until the hole is sideways.

This is closely related to how most people solve Rubik's cube. They know macros that fix very particular things, like, for example, a macro that can flip two particular edge cubies in place leaving all the rest of the cubies exactly the same as they were before the macro was applied. Suppose the macro you know flips the front-up and the left-up edge cubies in place, but to solve the particular jumbled cube you're holding, you need to flip edge cubies opposite each other on the bottom. You'd turn the cube over and then do two twists to put the cubies that need flipping in the front-up and left-up positions, apply the macro, and then undo the two twists. The two preparatory twists are like twisting the block sideways and putting it under the drill. Drilling the hole is like applying the macro, and undoing those two preparatory twists is like taking the block out of the drill press and setting it right-side-up again.

In mathematical terms, we can think of the movement of the block as a change of coordinates. If you think of the z-axis as pointing straight up, then putting the block under the drill is equivalent to, say, moving the block's x-axis to the z-axis of the world. This is a change of coordinates, and hence the title of this section.

On the cube these preparatory operations are almost always three or fewer twists. The macros can be complicated, but they're worth memorizing since you only need a small number of them.

If we use the letter M to represent the permutation corresponding to the macro and we represent by P the operation that twists the faces so that the ones you want to operate upon are set up for M, then the inverse of P, P^{-1} is the operation to restore the cubies that are not affected by M to their initial conditions. You would write the entire operation together as PMP^{-1}. In mathematics, this is called "the conjugation of M by P". The result: PMP^{-1}, is called a conjugate of M.

You will see this form over and over in books on group theory, but sometimes in the opposite form: $P^{-1}MP$. This opposite form is effectively just thinking of the operation from the macro's point of view—the cube was initially moved to the wrong configuration, so to set up, you have to undo the moves that made it wrong. Then the operation is performed followed by the moves to put it back in the wrong position ("wrong" only from the point of view of the macro, of course).

9.1 Change of Coordinates Exercises

Although it is not difficult, this is perhaps the most important single idea that you need to use to solve a jumbled cube. If you'd like to be sure you understand it, do the following exercise with the **Rubik** program.

When the program starts up, you will see "Flip UF, UL" in the "Defined Macros" choice area and a solved cube will be displayed. Click once on the "Flip UF, UL" area and those two cubies, the up-front and the up-left, will be flipped in place. Click again, and the cube will be restored to solved. The idea is that the macro does a very precise thing—it flips the two cubies in those slots, no matter what they are. Click on the up-face to turn it a quarter turn clockwise and again click on the macro. Again, it flips the two cubies currently in the up-front and up-right locations in place no matter what colors they are and leaves the rest of the cube unaffected.

Click again to put them back, and then turn the top face back to solved (or just click the Reset button).

If you want to see the macro that does the two flips in action, notice that as soon as you clicked on it, its macro pattern appeared in the "Current Macro" area. If you click in that area and then press the **return** key on your keyboard, you can watch **Rubik** run through the steps and convince yourself that the macro will really work.

Now, use the "Input Cube" command under the "Edit" pull-down menu to bring up the cubie editor. Use the editor to change the colors of the up-front and up-back cubies so that they are flipped. (If you're using the default coloration, cube U8 should be red, cube F2 white, cube F8 yellow and cube D2 red. Click on the Finish button and **Rubik** should display a cube with those two edge cubies flipped. (Remember that you can look at the rear view of the cube to convince yourself that you've got it right.)

Now, suppose you had somehow gotten the cube to this condition, almost solved and you need to finish the job. You know how to flip the UF and UL cubies in place, but only one of them is right. But if you give the back face a clockwise turn and then the left face another clockwise turn, the two bad cubies are in a position where the macro can operate on them. Apply the macro and then undo the preparatory setup by turning the left face counterclockwise and then the back face counterclockwise. Thus $P =$ BL so $P^{-1} =$ lb, and the M is the "Flip UF, UL" macro: $M =$ FRBLUlUbrfluLu.

By the way, here's an easy way to set up the cube for this example that obviates the use of the "Input Cube" command: From a solved cube, click on "Flip UF, UL", then click on the up-face to give it a quarter-turn. Apply the "Flip UF, UL" macro again, and twist the up-face back to its original position.

Do you see why this works? This is almost like the PMP^{-1} idea except that we applied an M^{-1} at the end. (The P is the "Flip UF, UL" macro and since the order of that macro is 2, P^{-1} is the same as P. The M is the quarter-turn of the top face.) The total operation is thus $PMP^{-1}M^{-1}$ which is called a commutator and leads us smoothly into the next topic.

10 Commutators

If A and B are two elements of a group (are two permutations, for example), then the commutator of A and B, sometimes written "$[A, B]$" is defined to be $ABA^{-1}B^{-1}$. More often you'll see it defined as $A^{-1}B^{-1}AB$, but the difference is unimportant since it just reverses the roles of each permutation with its inverse.

Why are they useful, and why would anyone ever have come up with this concept in the first place?

If this is the first time you have ever looked at mathematical objects like groups, it may be the first time you've ever run across a system where the main operation is not commutative. A system (including a group) where the order of operation does not matter is called commutative. In other words, a group is commutative if for every two elements a and b in the group, $a{*}b = b{*}a$. As we've noticed, this is certainly not the case with $\mathcal{R}$.

In a commutative system, the commutator of a and b would be $aba^{-1}b^{-1}$, but since the order of operations in a commutative system is unimportant, we can reverse the order of the middle two objects:

$$aba^{-1}b^{-1} = a(ba^{-1})b^{-1} = (aa^{-1}b)b^{-1} = (aa^{-1})(bb^{-1}) = ee = e,$$

where e is the identity element of the group. Thus in any commutative group, the commutator of any two objects is simply the identity, and if we were talking about permutation groups, then the commutator would not move any objects.

Even in $\mathcal{R}$, although most pairs of operations do not commute, there are some that do. Any operation commutes with itself, for example, or U and D also commute since they move completely different sets of cubies.

So for any two permutations in a group, if their commutator is the identity, those permutations commute. But if they don't commute, we can think of the commutator as a sort of "measure" of how non-commutative they are. We will see that if two permutations "almost commute" then their commutator is relatively "simple". If they are far from commuting, their commutator will be complex. Of course the terms "almost commute" and "simple" in this paragraph are not particularly mathematical topics. (Do not confuse this use of "simple" with "simple group". The word "simple" is used in this paragraph in a completely non-rigorous manner.)

Here's an example. In cycle notation, let $a = (1\ 2\ 3\ 4\ 5\ 6)(7\ 8\ 9)(10\ 11\ 12\ 13\ 14)$ and $b = (9\ 7)(15\ 16\ 17\ 18\ 19\ 20)$. Although both of them move a lot of elements, if you work it out, the commutator $[a, b] = (7\ 9\ 8)$. If you think about it, it should be clear why this happens. Although both permutations move a lot of objects, the only ones that are involved in common cycles in the two permutations are 7, 8 and 9. The permutation a moves, for example, 1, 2, 3, 4, 5 and 6 one step forward in a cycle, but since b does not move any of them, they are left in place, so the a^{-1} in the commutator undoes the action of a on all six of those elements.

Notice that we cannot say that two permutations are almost commutative if they only move a small number of objects in common. In the example below, only the object 6 is moved by the two permutations, but their product actually tangles up all the elements that either moves into one giant cycle:

$$(1\ 2\ 3\ 4\ 5\ 6)(6\ 7\ 8\ 9\ 10) = (1\ 2\ 3\ 4\ 5\ 7\ 8\ 9\ 10\ 6).$$

Also notice that two permutations *may* be completely commutative even if they both move all of the elements:

$$(1\ 2)(3\ 4) * (1\ 3)(2\ 4) = (1\ 4)(2\ 3) = (1\ 3)(2\ 4) * (1\ 2)(3\ 4).$$

The idea that permutation “almost commute” when they do not move many elements in common is only a rule of thumb that can, at times, be completely incorrect.

Let’s look at a practical use of the commutator concept to build a very useful macro: the “Flip UF UL” macro we used in Section 9.1.

Here is the strategy: We will find a series of cube moves that leaves the top face completely unchanged except that a single edge cubie on it is flipped. This is easier than it sounds—although we have to be careful with the top layer, our operation can arbitrarily trash the lower two levels.

If we have such a macro M, we’ll apply it to flip the one cubie on the top. Then we’ll rotate the top to put a different cubie in that location, at which point we will undo M by applying M^{-1}. This will obviously undo all the damage on the lower two layers and flip only the one cubie on the top layer. But we moved a different cubie into that location by twisting the top face, so a different cubie will be “unflipped”. After this, we undo the rotation on the top face and we’re done.

The only slightly tricky part is to obtain the macro M.

You’re welcome to look for your own scheme to find this M (it’s probably easier to search with the virtual cube, since you’re almost certain to mess up a physical cube in a search like this). Also, you might consider turning on the “Record” feature in **Rubik** so you don’t have to write down the moves as you make them. Or, once you’ve got a macro that works, use the “Undo” operation to figure out exactly what you did.

Here was my approach: I want to flip the UF cubie in place but other than that, I want to leave the rest of the U face exactly as it was. (You may wish to follow along with a virtual (or physical, if you like) cube. First, let’s move the six upper-left and upper right cubies out of the way with Rl. This puts all six of them on the back of the cube from me.

Next, I turn the front face around 180° with FF. This puts the cubie I want to flip in the FD location with what was originally the up-face pointing down. I’d like to rotate the bottom face, but some of the cubies that were originally on top would be moved, so I’ll bring them back to the top (which won’t be affected by a turn of the bottom) with Lr.

At this point, the top is pretty pure, but we’ve got the wrong cubie in the UF position. The correct cubie is in the FD position so if I turn around the front face again I could put it in place, but it wouldn’t be flipped, and I’d move two of the top pieces. Here’s the trick: rotate the bottom face (it doesn’t matter which way, but let’s go counter-clockwise for an d move so we can still see the cubie that interests us in the LB location. If we do a Rl move those same six top cubies are protected, and a F move puts the cubie (now flipped) in the correct place next to its unflipped neighbors.

Finally, a rL gets us exactly the macro we need. If you put all those moves together, you'll get the 12-move macro $M =$ RlFFLrdRlFLr.

Type those twelve steps into the "Current Macro" window, reset the cube, and click on the Apply Macro button to make sure you've recorded it correctly. If you did, you'll get the top face with just the UF cubie flipped and a bunch of damage down below. The grand goal, of course, is to get the "Flip UF, UR" macro but this should just be MuM^{-1}U.

Assuming you've still got the macro M in the "Current Macro" window, reset the cube, click Apply Macro, then do u, then hold down the **shift** key and click Apply Macro again followed by a U. Remember that the **shift** key causes the inverse operation to be performed. This should be exactly what you want. To write it out completely, we just need to work out the inverse of M, and the whole macro that flips UF and UR in place is:

RlFFLrdRlFLruRlfLrDRlffLrU

The macro that **Rubik** uses for the same thing, FRBLUlUbrfluLu, is shorter, but probably not as easy to remember[3]. Although the derivation our derivation of the macro above may seem a little obscure, there is a way to think of it in terms of slice moves that is quite intuitive.

Here is effectively the same macro, but with slices instead:

*LDD*Rd*LD*R

Type the macro above into the "Current Macro" window, reset the cube, and then step through it as you read this text by pressing the **right-arrow** key on the keyboard.

The first move gets the piece you want to flip on the bottom of the cube leaving the other six up-pieces on top. Next, twist the bottom 180° to get the desired cubie out of the way. The 180° is good, since when you reverse that first slice move, it flips the cubie that you eventually want in the UF position to the correct orientation.

Next, you are going to drive the same middle slice down again and rotate the cubie into that slot, but if you just do the rotation it will flip that cubie back into the wrong orientation. Thus, before the slice move you need to get the cubie out of the way which can be done with a quarter turn of the bottom in either direction (counter-clockwise was chosen for this example so you can keep your eye on the cubie you're trying to flip).

That quarter turn is followed by the same slice move you used originally after which you can rotate the now-flipped cubie into its slot and reverse the slice move to return it to the top.

Textual descriptions like this are often difficult to follow, but you can reset the cube, and single step the macro again and again until you have an intuitive visual feeling for what is going on. When the single stepping reaches the end of the macro, the pointer is reset to the beginning of the macro. (If you then click on the Reset button, you'll need to click in the "Current Menu" area to return **Rubik**'s attention there.) You can back up in single steps by pressing the **left-arrow** key.

[3] See Section 11 for information about how the macros used by **Rubik** were obtained.

As a cube-solving macro, the author actually uses a reversed form of this one. It seems quicker to "push" the slice away than to "pull" it toward you, so what the author does is the exact same movements except that he is effectively standing facing the back face of the cube. You can try both and decide for yourself.

Another interesting thing to notice is that if you repeat the macro twice, it cycles three of the cubies on the bottom of the cube. Try executing it twice to see what is meant[4]. With the slice moves, this is an easy-to-remember 16-move macro (if a slice move counts as a single move) to cycle three slices as well. But really it is only 14 moves, since you'll note that when the macro is applied twice in a row the last and first moves in the middle cancel out. But when you do that, you notice that your sequence includes DDD which can be replaced by d, so it really only contains 12 moves! Here is the condensed 12-move version:

*LDD*Rd*Ld*Rd*LD*R

Going back to the original problem—to find a macro that flips to top edge cubies in place without affecting the rest of the cube, one of the reasons this particular commutator works so well is that the operations you used affected such different parts of the cube. One part of the commutator simply rotated the top face; the other flipped only one cubie in the top face and did all its damage to the rest of the cube. As you can see by the result, those two operations, although they do not commute, are very close to commuting and hence have a simple commutator.

As an interesting exercise, see if you can find a macro in the form of a commutator that twists one corner cubie clockwise and another counter-clockwise but leaves the rest of the cube as it was. Hint: you need to find a macro that twists one corner of the top face and does not affect any other cubies on that face. The answer appears at the end of the next section.

10.1 Commutators for Cycling Cubies

We know from our analysis in Section 8 that it is impossible to find a macro that will exchange two corner cubies (or two edge cubies) that will not move any other cubies. But it is possible to find a macro that will move three cubies in a cycle like (UF UB DF) or (URB UFR ULF).

We'll begin with a method to construct a cycle of three corner cubies using roughly the same strategy as before. We'll find a sequence of moves to swap two adjacent corner cubies on the top face leaving the rest of the top face intact. Then we'll rotate the top face a quarter turn and undo what we just did. This will re-swap one of the cubes we already swapped, but will undo all the rest of the cube damage. The net result will be a cycle of three corner cubies since the permutation structure will look like this: $(1\ 2)(2\ 3) = (1\ 3\ 2)$.

It is quite easy to find a permutation that swaps two cubies on the up face while leaving the rest of that face intact: LrDRdl, whose inverse is LDrdRl. Thus the final permutation that cycles three corner cubies and does nothing else is LrDRdlULDrdRlu.

We can do this with a *very* simple macro that is constructed as a commutator, but based on

[4] You could easily have discovered this yourself since you could learn that the order of your macro is 6, and thus it might be a good idea to try it 2 or 3 times to see what resulted

the following observation about a product of permutations:

$$(A\ B\ C\ D)(A\ C\ D\ B) = (B\ D\ C).$$

It is a bit more difficult to analyze since it involves slice moves that move the cubies that are being exchanged, so it's a little difficult to keep track of exactly what cubies are being cycled. In this case it's easiest just to do the operation to a cube and see what happens. The permutation product above amounts to cycling four cubies, reversing two of them and inverting that cycle.

A single slice move, *L, cycles the face cubies as follows: (UF DF DB UB). If we then rotate the up-face by $180°$ we effectively swap the cubies that were UB and DB, so the inverse of *L produces the permutation (UF DB UB DF). The product (UF DF DB UB)(UF DB UB DF) = (DF UB DB). A further turn of the up face by $180°$ returns everything else to its original locations. In the previous example, we left the cubies on the top face fixed and trashed the rest of the cube; in this case we effectively left the slice in good shape and again, trashed the rest of the cube. The full macro to achieve (DB UF UB) is thus *LUU*RUU, and this provides a method to cycle three cubies. This is a commutator since the inverse of *L is *R and the inverse of UU is UU.

The solution to the problem stated earlier to find a macro that will rotate two corner cubies in place and will not affect any of the other cubies is based on this operation that rotates a single corner cubie: LdlfdF. Then rotate the top face, undo the operation, and rotate the top face back. An answer (and there are, of course, an infinite number of others) is: LdlfdFUfDFLDlu.

10.2 Finding Your Own Commutators

As we saw above, it is nice to have as one of the elements of a useful commutator an operation that does a very simple thing, at least relative to some subset of the cubies on the cube. This can then be combined as a commutator with other operations to possibly form useful macros for cube solution.

It is also nice, of course if the pieces of the commutator are short.

Here are a couple of useful building blocks for commutators. It's your job to find operations which, when combined with them, do useful work on the cube.

- FUdLLUUDDRU. This operation is called the monoflip and it flips exactly one cubie on the top face.
- rDRFDf. This is the monotwist that twists one cubie on a face.
- FF. This is the monoswap that swaps a pair of edges in a slice.
- rDR. This cycles three corners, but is not quite as useful.

11 Using "Solve" to Find Good Macros

An interesting but very difficult problem is to determine, from any given position, the minimum number of moves required to convert that position back to a solved cube. The first question that arises is what is meant by the word "minimum"? In other words, what counts as one move? Most of the work that has been done in this area is based on two definitions of a move. One is called the "quarter-turn metric" and the other, the "half-turn metric". The quarter-turn metric counts as a move any single rotation clockwise or counter-clockwise of $90°$. The half-turn metric allows a half-turn as a single move as well.

The "distance" between any two positions is the minimum number of moves required to convert one to the other, counted in one of the two metrics. Obviously, the distance in the quarter-turn metric is at least as large as the distance in the half-turn metric, since any half-turns would have to be replaced by a pair of quarter turns. Or another way to think of it is that any solution in the quarter-turn metric is also a solution in the half-turn metric, although there are additional possibilities in the half-turn metric than may (and usually do) provide a shorter sequence of moves.

The **Rubik** program currently measures distances in the quarter-turn metric. In fact, if you look in the little window above the "Current Macro" window labeled "Macro length", the number in that window represents the number of quarter-turns required to apply the macro in the "Current Macro" window.

A related question is this: what is the "worst" possible jumbling of a cube? In other words, what position or positions require the most moves to return them to solved? This maximum size of the minimum solution is sometimes called "God's number". God's number is not known, but there are known bounds on it.

In the quarter-turn metric, it is known that God's number has to be 24 or more. The position called "superflip" where all the edge cubies are flipped in place is known to require 24 moves to solve it. (This was determined by exhaustive search on a computer.) God's number may, in fact, be 24 since nobody has ever found a position that is known to require more than 24 moves.

It is possible to obtain a crude lower bound for God's number with the following observation. From a solved cube, there are at most 12 possible arrangements after the first twist (six faces, clockwise or counter-clockwise). After two moves, there are at most $12 \cdot 11$ positions, not 12^2 since one of the 12 moves undoes the original. After three, there are at most $12 \cdot 11^2$ positions, and following the same reasoning, after n moves, there are at most $12 \cdot 11^{n-1}$ possible positions. Since it takes as many moves to solve a position as to get to that position from solved, n must be at least large enough that $12 \cdot 11^{n-1} > 4.32 \cdot 10^{19}$ where the second number is the total number of cube positions. Solving for n tells us that the lower bound must be at least 19. Slightly more careful calculations (throwing out moves like FFF or FBfb, for example), one can show that God's number must be at least 21.

There are computer programs that can find the minimum number of steps from any given position to solved, but for each position, they usually require a day or so of computation, and in bad cases may require months.

The Solve button in **Rubik** does not find the minimum solution; it only finds one that is not too long. It looks for a while and then prints the best solution it has found up to that point. Depending on your computer speed and how much patience you have, you can change what is meant by "a while". In the "Edit" pull-down menu, you can set the time spent in a search for a good solution to either "Long" or "Very Long".

Although its solutions are not guaranteed to be the best, at least they are usually not too long, so you can use the program to search for macros.

For example, in Section 10 we found a macro to flip the UF and UL edge cubies in place that is 26 moves long. If you use the “Input Cube” command to input a cube with just those two cubies flipped and ask **Rubik** to solve it, it comes up with a much shorter (14 move) solution: FRBLUlUbrfluLu. Since this is the method to get from the flipped cubies to solved, you can invert it to UlULFRBuLulbrf to obtain a macro that goes from solved to the two flipped cubies configuration. (Actually, in a special case like this, the macro has order two, so the sequence works either forward or inverted. Generally, however, the solution **Rubik** finds needs to be inverted, and it never hurts to do so.)

Obviously this strategy can be applied to find a sequence for any legal macro you’d like to use. Just set up the situation on an otherwise-solved cube that you’d like to convert to solved, have **Rubik** find a set of moves that solves it, and invert those moves. Usually the macros it finds will not be too bad. For example, the optimal solution for “superflip” requires 24 moves and **Rubik** finds a 26 move solution fairly quickly.

On the other hand, **Rubik**-generated macros may be difficult to memorize for use on a physical cube. For example, **Rubik**’s 26-move superflip is FLULbudLfubRlBFFudFFBBRRud (or its inverse, since it is order-2). But here is a 36-move superflip that is almost trivial to memorize and execute: 2(4(*RU)>R>D)4(*RU). (The number 36 does not include the whole-cube moves in the turn count, but counts a slice move as two quarter turns, which it really is.) You can do this one with your eyes closed: “slice-up-slice-up-slice-up-slice-up-turn cube” and repeat that sequence two more times. The basic pattern: 4(*RU)>R>U makes a pretty nice pattern to run in demo mode. Just type it into the “Current Macro” window and press the Start Demo button. If you find those whole-cube moves annoying in demo mode, just take them out and use 4(*RU)4(*DF)4(*BL).

12 How Humans (Even You!) Can Solve the Cube

With patience, you can solve the cube with just 5 macros: one to flip two particular edge cubes, one to rotate two particular corner cubies, one to cycle three edge cubies, one to cycle three corner cubies, and one to swap two edge cubies and two corner cubies.

From any jumbled cube, first get all of the cubies in their correct positions (although possibly flipped and rotated). You will almost certainly have to use the idea of changing coordinates discussed in Section 9.1 to put the cubies you want to cycle or exchange in the proper places with one or two twists that you can undo after you’ve applied the macro.

If you only know how to swap a particular pair of edge and corner cubies at the same time, it may require a number of preparatory moves to get all four of them in the right positions, so if you’ve got everything right except for a pair of edge and corner cubies, it’s probably easiest to get two corners in the right places and exchange them and then the edge repairs can be done with one or two of the “cycle three edges” commands.

Once all the cubies are in their correct locations, you’ll need to twist and flip some of them. Use the same strategy you used before—a twist or two will put a pair that need flipping or rotating in the right places after which you apply the macro and undo the preparatory twists.

This method will obviously work but it will take a long time. The reason is that you are wasting a huge amount of effort at the beginning by using extremely restrictive macros that move only a tiny number of cubies when in reality you don't care what happens to the cubies that are out of place anyway. Think about putting the first corner in place. You can probably do that with just a move or two and the second corner won't be much harder.

People who are good at solving the cube have a whole set of more and more restrictive macros that get the pieces into position and then they have a set like the five listed above to do the final work.

Most people tend to solve the cube by first getting the top layer right, then the middle layer, and finally the bottom. Although this is very straight-forward, it has the disadvantage that once the top cubies are all in place, almost any twist involves that top layer, so some damage will be done that needs to be repaired.

The world champion speed-demons often use a different method that avoids this problem. First they solve a 2×2 corner of the cube, at which point there are three sides that can be turned freely with no effect on the solved portion. They next extend that to a $2 \times 2 \times 3$ block which still leaves two faces that can be turned without affecting the solved portion, and a lot of useful work can be accomplished just turning those two faces.

If you search on the internet, it's easy to find dozens of descriptions of useful cube-solving macros. A collection of a few useful macros can be found in Appendix B but you'll have more fun if you try to work them out yourself.

You can practice setting up for macros and then undoing the setup easily on the virtual cube. First learn exactly what cubies are affected by the half-dozen or so built-in macros. Then do the twist or two for the setup of each one, but rather than apply the macro by hand, just click on it in the window to have it happen instantly. Undo your setup moves and continue. After you are confident about your use of the macros, you can learn to do them as well, and you'll be in a position to solve a jumbled cube without the help of the **Rubik** program (or a screwdriver).

13 Subgroups of the Cube Group $\mathcal{R}$

In Section 7 we took a cursory look at some of the subgroups of $\mathcal{R}$. In this section we'll look at more examples and in addition we'll learn something about group generators and Cayley graphs.

13.1 Group Generators

Given a group $\mathcal{G}$ (and in this section, we'll almost always use Rubik's group $\mathcal{R}$ as our group), then if $S \subset \mathcal{G}$ is any subset of the group, then the subgroup $\mathcal{H}$ generated by S is the smallest subgroup of $\mathcal{G}$ that contains all the elements of S.

Surely such a subgroup exists. If any collection of different subgroups contain all the members of S, then their intersection (which is also guaranteed to be a subgroup) also contains all the elements of S and is contained in all of them. The group $\mathcal{G}$ is a subgroup of itself, so there is at least one group in that intersection.

Intuitively, the subgroup generated by S is the collection of all the elements you can get to by repeatedly multiplying members of S or their inverses together. In the case of $\mathcal{R}$, it's the set of all positions you can arrive at, starting with the jumbled cube and applying only that subset of moves in any order.

For example, the subgroup of $\mathcal{R}$ generated by $\{$ F $\}$ contains four members. If you're only allowed to turn the right face, you can only get to four different cube configurations. The subgroup generated by $\{$ FF $\}$ is even smaller: two positions. In other words, if you're only allowed to make half-turns of one face, there are only two possible positions you can achieve starting from solved.

Obviously, the group generated by $\{$ F, B, R, L, U, D $\}$ is the entire group $\mathcal{R}$—just a small number of generators can generate a huge group. We are interested, of course, in subgroups that fall between the extremes mentioned in this and the previous paragraph.

13.1.1 Cyclic groups

The simplest situation, of course, is subgroups generated by a single element g. If there is some n such that $g^n = 1$, then the entire group is given by $\{1, g, g^2, g^3, \ldots, g^{n-1}\}$. (For infinite groups, which we're pretty much ignoring here, it may be that g^n is never equal to 1, so the group generated by such a g would consist of $\{\ldots g^{-2}, g^{-1}, 1, g^1, g^2, \ldots\}$.) Such groups with a single generator are called cyclic groups (even if they're infinite). All cyclic groups of the same order are isomorphic, so, for example, the group generated by F and the group generated by R behave in essentially the same way.

Since the Rubik group $\mathcal{R}$ is finite, any single element of $\mathcal{R}$ generates a cyclic subgroup whose size simply depends on the order of that element. Using **Rubik** it is easy to find the orders of group elements; just type them into the "Current Macro" input area and click on the $\boxed{\text{Macro Order}}$ button. If you type an "F" into that input area, the $\boxed{\text{Macro Order}}$ command will tell you it is 4, as you would suspect, but it is easy to try other random (or non-random, of course) combinations to see what their orders are.

The simple FR has a (perhaps surprisingly large) order of 105, meaning that you would have to repeat that two-turn combination 105 total times before a solved cube would return to solved. Try some experiments, especially with commutators. Use **Rubik** to find the order of RUUdBd. Can you find any elements with a larger order?

As you'd expect (I hope) changing coordinates will not affect a macro's order. In other words, if the order of P is n, and if Q is any other macro, then the order of QPQ^{-1} is also n. Do you see why? Can you prove it?

The order of an element (and hence the order of the cyclic group it generates) has to divide the order of the entire group which we showed earlier to be:

$$43252003274489856000 = 2^{27} \cdot 3^{14} \cdot 5^3 \cdot 7^2 \cdot 11.$$

Thus you're never going to find any elements whose order is divisible by 13, for example. All their orders must have prime factors among those shown above.

There are some interesting theorems in group theory called the Sylow theorems that tell us a bit more about the structure of finite groups. We will not state or prove those theorems (you can find them in any elementary text on group theory), but will simply state that one of them guarantees that there is at least one subgroup of order 11 of $\mathcal{R}$. How could we find such an example?

One easy way would be with **Rubik**. Construct a cube that moves 11 of the 12 edge cubies in a cycle. We can do that with the "Input Cube" command. Then click on the "Solve" button and we will be presented (after a short wait) with a macro that undoes this 11-cubie cycle. The inverse of this will take a solved cubie to a position that is the first step in an 11-cycle, so the inverse of the macro given to us by "Solve" will do the trick. In fact, we don't even have to take that inverse—the order of an element is the same as the order of its inverse, so we can directly use the result given to us by **Rubik**. Here's one such example (by no means guaranteed to be the shortest such example): ruFBufDBUDbuRRdLLuLLdLLuRR.

See if you can find your own.

By the way, you may come up with a macro of order 22 instead of 11 because the way you cycled the cubies, after 11 steps some of them are flipped, so 22 steps are required to complete the loop. In this case, the square of the result will have order 11.

Do you see how you might construct other cyclic groups of various orders? Can you construct an element of order 55 this way?

13.1.2 The subgroup generated by FF and RR

For our first example of a non-cyclic group defined in terms of generators, et's begin with a simple example that we have examined a bit already: the group generated by $\{$ FF, RR $\}$. (Remember that both moves are considered as units—every move of the right face *has* to be a half-turn and similarly for the front face. In fact, to emphasize that, let's give single letter names to each: $\phi = \mathsf{FF}$ and $\rho = \mathsf{RR}$.

We know that $\phi^2 = \rho^2 = 1$ and we found earlier (or we can easily check with **Rubik**) that $(\phi\rho)^6 = (\rho\phi)^6 = 1$ and that no smaller power will do. In other words, the order of $\phi\rho$ and also of $\rho\phi$ is 6.

If we start looking for possible group members, it is clear that there are only the following 18 possibilities:

$$\begin{array}{cccccc}
1 & (\rho\phi) & (\rho\phi)^2 & (\rho\phi)^3 & (\rho\phi)^4 & (\rho\phi)^5 \\
\phi & \phi(\rho\phi) & \phi(\rho\phi)^2 & \phi(\rho\phi)^3 & \phi(\rho\phi)^4 & \phi(\rho\phi)^5 \\
\rho & (\rho\phi)\rho & (\rho\phi)^2\rho & (\rho\phi)^3\rho & (\rho\phi)^4\rho & (\rho\phi)^5\rho
\end{array}$$

The ϕs and ρs have to alternate or they will cancel to the identity. Furthermore, if there are more than 6 $(\phi\rho)$ or $(\rho\phi)$ pairs in a row, that set of 6 will also cancel to the identity. It is clear that if you multiply any of the elements in the list above by ϕ or by ρ on the right or left, it will cancel to something else on the list. Thus the size of the generated subgroup is at most 18.

But there are some duplicates in the list above. We will show one example of a duplicate pair; there are six similar duplicate pairs in the list above, which will show that the size of the generated subgroup is 12 (and you can test this with **Rubik**, if you wish, to see that all 12 are different).

Here we will show that $\phi(\rho\phi)^2 = (\rho\phi)^3\rho$ because they are both inverses of $(\rho\phi)^3\rho$ and the inverse of a group element is unique. To see that they are both inverses, we just multiply them together and show that both collapse to the identity:

$$(\rho\phi)^3\rho \cdot \phi(\rho\phi)^2 = (\rho\phi)^3(\rho\phi)(\rho\phi)^2 = (\rho\phi)^6 = 1.$$

Similarly, because of repeated cancellations of equal elements in the center, we obtain:

$$\begin{aligned}
(\rho\phi)^3\rho \cdot (\rho\phi)^3\rho &= \rho\phi\rho\phi\rho\phi\rho\rho\phi\rho\phi\rho\phi\rho \\
&= \rho\phi\rho\phi\rho\phi\phi\rho\phi\rho\phi\rho \\
&= \rho\phi\rho\phi\rho\rho\phi\rho\phi\rho \\
&\quad \vdots \\
&= \rho\rho \;=\; 1.
\end{aligned}$$

The complete group consists of the members of the top two rows of the 18-element list. Or the top and bottom row; whichever you prefer—each element on the bottom row is the same as an element in the middle row. This group can be shown to be isomorphic to the dihedral group on six objects, which is essentially the set of symmetries of a regular hexagon that you're allowed to rotate or flip over.

In fact, a slightly easier to use set of representatives for the members of the group is this (the parentheses just help visualize the grouping). It's a good exercise to convince yourself that all the elements in the list below are different and to try to multiply together various combinations of them.

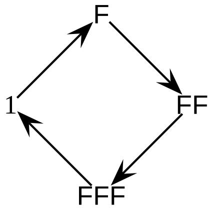
$$\begin{array}{lllllll}
1 & \rho & (\rho\phi) & (\rho\phi)\rho & (\rho\phi)^2 & (\rho\phi)^2\rho & \\
 & \phi & (\phi\rho) & (\phi\rho)\phi & (\phi\rho)^2 & (\phi\rho)^2\phi & (\phi\rho)^3 = (\rho\phi)^3
\end{array}$$

This sort of analysis is often possible given a set of relations that the generators satisfy, but it is often surprisingly difficult to do such an analysis. Who would guess, for example, the the group generated by $\{$ F, R $\}$ satisfying: $\mathsf{F}^4 = \mathsf{R}^4 = (\mathsf{FR})^{105} = 1$ (plus a few other relationships) would generate a subgroup containing 73483200 members?

The concept of generators, however, is very powerful when we are working on a puzzle like Rubik's cube. The generators are basically the set of moves we allow ourselves to do, and the size of the generated group is the number of positions achievable from that set of moves.

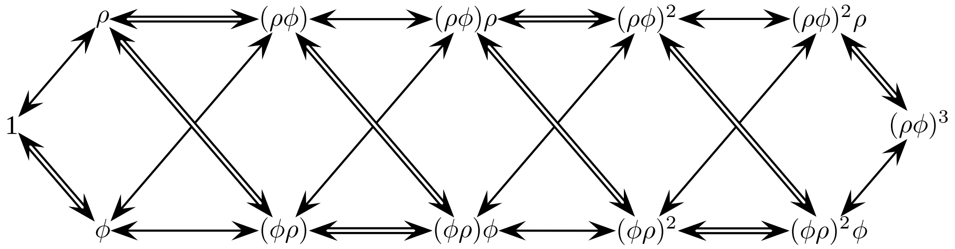
13.1.3 The Cayley Graph

One nice way to visualize how a group is generated from a set of generators is with a Cayley graph. A Cayley graph is simply a picture with nodes indicating each group element and arrows from one to the next when one of the generators will take you from that element to the next. As an almost trivial example, here is the Cayley graph for the group generated by the single element F:



The Cayley graph for the group examined in the last section and generated by ρ and ϕ is a little more interesting. Starting from 1, any element can be obtained from a previously-obtained element by multiplying it on the left or right by ϕ or ρ. Obviously, at least some of the time this will cancel a ϕ or a ρ that was multiplied on earlier, but eventually you will obtain a complete list of the elements in the group.

The figure below illustrates the Cayley graph for that group. Elements that can be obtained from another by multiplying by ϕ are connected with simple arrows; elements that can be obtained from another via a multiplication by ρ are connected with double-line arrows.



Let's examine a more complicated situation: A_4, the alternating group on four objects which was mentioned in Section 8.1. As a reminder, the alternating group on n elements is the permutation group consisting of all even permutations of those objects. An even permutation is a permutation that contains an even number of 2-cycles.

Here are the elements of A_4:

(1)	$(1\ 2\ 3)$	$(1\ 2\ 4)$	$(1\ 3\ 4)$	$(2\ 3\ 4)$	$(1\ 3\ 2)$
$(1\ 4\ 2)$	$(1\ 4\ 3)$	$(2\ 4\ 3)$	$((1\ 2)(3\ 4)$	$(1\ 3)(2\ 4)$	$(1\ 4)(2\ 3)$

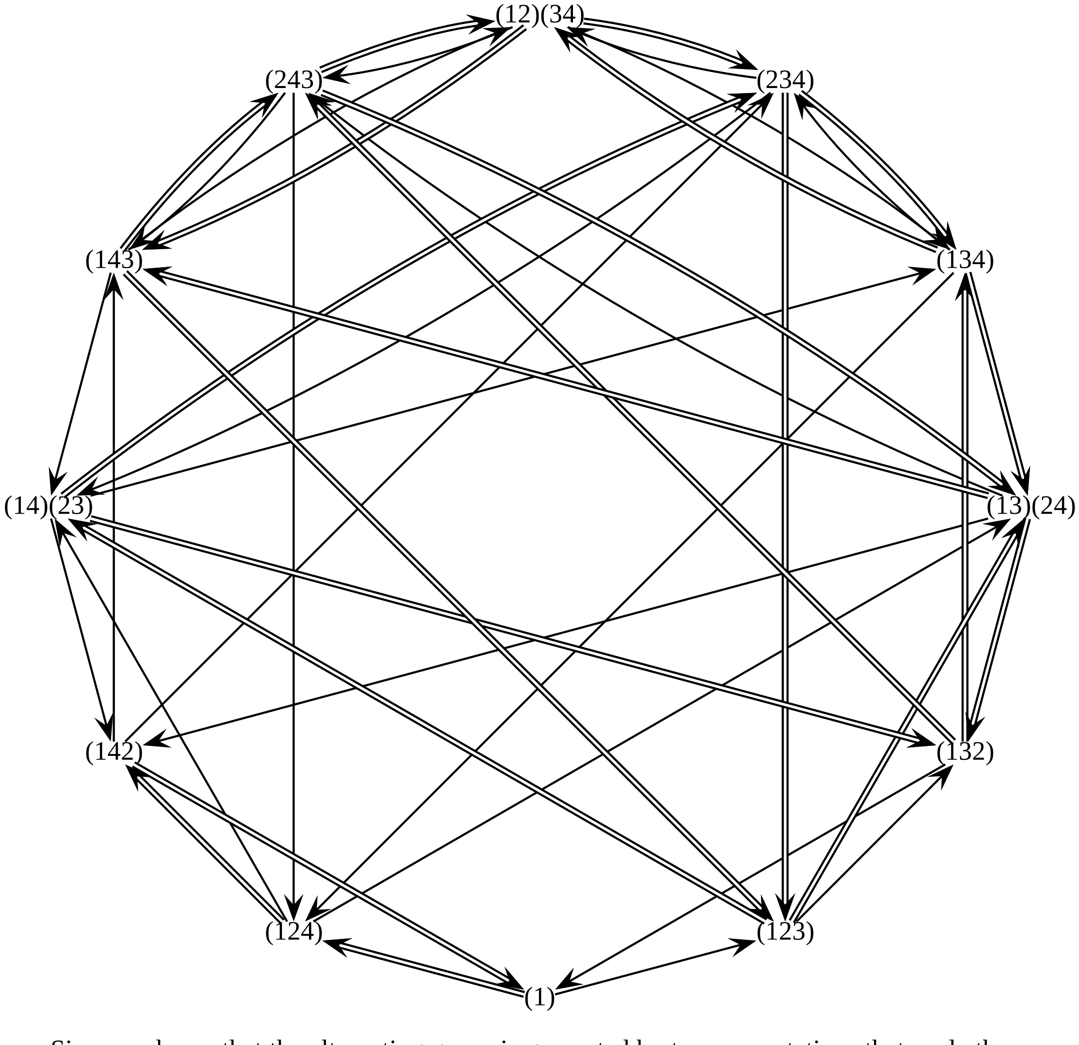
Table 3 is the multiplication table for the alternating group A_4.

	(1)	(123)	(124)	(134)	(234)	(132)	(142)	(143)	(243)	(12)(34)	(13)(24)	(14)(23)
(1)	(1)	(123)	(124)	(134)	(234)	(132)	(142)	(143)	(243)	(12)(34)	(13)(24)	(14)(23)
(123)	(123)	(132)	(13)(24)	(234)	(12)(34)	(1)	(143)	(14)(23)	(124)	(134)	(243)	(142)
(124)	(124)	(14)(23)	(142)	(13)(24)	(123)	(134)	(1)	(243)	(12)(34)	(143)	(132)	(234)
(134)	(134)	(124)	(12)(34)	(143)	(13)(24)	(14)(23)	(234)	(1)	(132)	(123)	(142)	(243)
(234)	(234)	(13)(24)	(134)	(14)(23)	(243)	(142)	(12)(34)	(123)	(1)	(132)	(143)	(124)
(132)	(132)	(1)	(243)	(12)(34)	(134)	(123)	(14)(23)	(142)	(13)(24)	(234)	(124)	(143)
(142)	(142)	(234)	(1)	(132)	(14)(23)	(13)(24)	(124)	(12)(34)	(143)	(243)	(134)	(123)
(143)	(143)	(12)(34)	(123)	(1)	(142)	(243)	(13)(24)	(134)	(14)(23)	(124)	(234)	(132)
(243)	(243)	(143)	(14)(23)	(124)	(1)	(12)(34)	(132)	(13)(24)	(234)	(142)	(123)	(134)
(12)(34)	(12)(34)	(243)	(234)	(142)	(124)	(143)	(134)	(132)	(123)	(1)	(14)(23)	(13)(24)
(13)(24)	(13)(24)	(142)	(143)	(243)	(132)	(234)	(123)	(124)	(134)	(14)(23)	(1)	(12)(34)
(14)(23)	(14)(23)	(134)	(132)	(123)	(143)	(124)	(243)	(234)	(142)	(13)(24)	(12)(34)	(1)

Table 3: The Alternating Group A_4

What we will examine here is the Cayley graph of the alternating group A_4 based on two permutations that generate the entire group: $(1\ 2\ 3)$ and $(1\ 2\ 4)$.

The following figure shows the Cayley graph with the following conventions. If a group element x can be obtained from an element y by pre- or post-multiplying it by $(1\ 2\ 3)$ then a simple arrow points from x to y. If y can be obtained from x by pre- or post-multiplication by the permutation $(1\ 2\ 4)$ then a double-line arrow points from x to y.



Since we know that the alternating group is generated by two permutations that cycle three objects, two of which are the same, it is actually easy to find a subgroup of the Rubik cube group $\mathcal{R}$ that is isomorphic to A_4. Just take two permutations that cycle three edge cubies (or corner cubies, it doesn't really matter), and as long as two of those cubies are shared, those two operations will generate a subgroup of $\mathcal{R}$ isomorphic to A_4.

Here are two moves that do the trick: rdlFFLDRUFFu, UdLuDFFUdLuD, although there are dozens of other pairs that would generate similar isomorphic subgroups.

13.1.4 More subgroups of $\mathcal{R}$

It is surprisingly difficult to find small subgroups based on simple sets of generators. The sizes of the subgroups seem get large fairly rapidly except in the simplest cases. It is certainly possible to construct small subgroups but they are usually based on fairly complex generators. In Section 13.1.1, for example, we found a single generator that produced a group of order 11, but it was based on a generator that is 27 moves long. (Well, **Rubik** found a 27 move sequence; there may be shorter ones.)

Listed below are some subgroups that are generated from a small number of generators. As you can see, most of them would not be too useful for learning how to manipulate the full cube.

	Generators	Size	Factorization
1	U	4	2^2
2	U, RR	14400	$2^6 \cdot 3^2 \cdot 5^2$
3	U, R	73483200	$2^6 \cdot 3^8 \cdot 5^2 \cdot 7$
4	RRLL, UUDD, FFBB	8	2^3
5	Rl, Ud, Fb	768	$2^8 \cdot 3$
6	RL, UD, FB	6144	$2^{11} \cdot 3$
7	FF, RR	12	$2 \cdot 3^2$
8	FF, RR, LL	96	$2^5 \cdot 3$
9	FF, RR, LL, BB	192	$2^6 \cdot 3$
10	FF, RR, UU	2592	$2^5 \cdot 3^4$
11	FF, RR, LL, UU	165888	$2^{11} \cdot 3^4$
12	FF, BB, RR, LL, UU	663552	$2^{13} \cdot 3^4$
13	FF, BB, RR, LL, UU, DD	663552	$2^{13} \cdot 3^4$
14	LLUU	6	$2 \cdot 3^2$
15	LLUU, RRUU	48	$2^4 \cdot 3$
16	LLUU, FFUU	1296	$2^4 \cdot 3^4$
17	LLUU, FFUU, RRUU	82944	$2^{10} \cdot 3^4$
18	LLUU, FFUU, RRUU, BBUU	331776	$2^{12} \cdot 3^4$
19	LUlu, RUru	486	$2 \cdot 3^5$
20	LUlu, RUru, LDld	17496	$2^3 \cdot 3^7$
21	LUlu, RUru, LDld, RDrd	52488	$2^3 \cdot 3^8$
22	>F, >L	24	$2^3 \cdot 8$
23	*F, U	184320	$2^{12} \cdot 3^2 \cdot 5$
24	*F, U, *U	4423680	$2^{15} \cdot 3^3 \cdot 5$

In the table above, entry 4 is called the slice-squared group, entry 5 is the slice group, and entry 6 is the anti-slice group. Entries 19, 20 and 21 are produced by small sets of similar commutators. Below the double line, entry 22 is the whole-cube group and the other two are produced with a standard move of the up face combined with one or two slice moves.

Another interesting observation is that the same subgroup is generated by entries 12 and 13—adding the additional DD move did nothing. This indicates that a DD can be generated by the other 5 moves, and it can: DD $=$ RRFFBBLLUURRFFBBLL. Similarly, the entire cube group $\mathcal{R}$ can be generated by quarter-turns of only 5 faces:

D = RRLLUURRBBRRLLFFLLuRRLLUURRBBRRLLFFLLUU.

14 Direct Product Groups

Given two groups, $\mathcal{H}$ and $\mathcal{K}$, it is possible to construct another group $\mathcal{G}$ that is called the direct product of $\mathcal{H}$ and $\mathcal{K}$ (written $\mathcal{G} = \mathcal{H} \times \mathcal{K}$) as follows:

For every $h \in \mathcal{H}$ and $k \in \mathcal{K}$ then $(h, k) \in \mathcal{H} \times \mathcal{K}$, where (h, k) is an ordered pair with h the first element and k the second. Assuming that $*$ is the group operator for $\mathcal{H}$ and $\odot$ is the group operator for $\mathcal{K}$, then the group operator $\otimes$ for $\mathcal{G} = \mathcal{H} \times \mathcal{K}$ is defined as follows:

$$(h_1, k_1) \otimes (h_2, k_2) = (h_1 * h_2, k_1 \odot k_2).$$

In other words, just apply the operators of the appropriate groups to the elements of those groups.

Of course it needs to be proven that $\mathcal{H} \times \mathcal{K}$ is a group, so we need to show that it contains an identity, inverses, is closed, and satisfies the associative laws. All of these are petty straightforward and follow from similar properties of the two groups that make up the direct product:

Identity: If e_h and e_k are the identities of $\mathcal{H}$ and $\mathcal{K}$, respectively, then (e_h, e_k) is the identity of $\mathcal{H} \times \mathcal{K}$:

$$(e_h, e_k) \otimes (h, k) = (e_h * h, e_k \odot k) = (h, k).$$

Inverses: If $(h, k) \in \mathcal{H} \otimes \mathcal{K}$ then its inverse is (h^{-1}, k^{-1}), since:

$$(h, k) \otimes (h^{-1}, k^{-1}) = (h * h^{-1}, k \odot k^{-1}) = (e_h, e_k).$$

Closure: If $(h_1, k_1) \in \mathcal{H} \times \mathcal{K}$ and $(h_2, k_2) \in \mathcal{H} \times \mathcal{K}$ then h_1 and h_2 are in $\mathcal{H}$ and because of the closure of $\mathcal{H}$, so is $h_1 * h_2$. Similarly, $k_1 \odot k_2 \in \mathcal{K}$. Thus the operator $\otimes$ is well-defined:

$$(h_1, k_1) \otimes (h_2, k_2) = (h_1 * h_2, k_1 \odot k_2) \in \mathcal{H} \times \mathcal{K}$$

since $\mathcal{H} \times \mathcal{K}$ contains *all* possible ordered pairs of elements from $\mathcal{H}$ and $\mathcal{K}$.

Associativity: As with the other properties, the associativity of the operation $\otimes$ follows directly from the associativity of the operations $*$ and $\odot$:

$$\begin{aligned}(h_1, k_1) \otimes ((h_2, k_2) \otimes (h_3, k_3)) &= (h_1, k_1) \otimes (h_2 * h_3, k_2 \odot k_3) \\ &= (h_1 * (h_2 * h_3), k_1 \odot (k_2 \odot k_3)) \\ &= ((h_1 * h_2) * h_3, (k_1 \odot k_2) \odot k_3) \\ &= ((h_1 * h_2), (k_1 \odot k_2)) \otimes (h_3, k_3) \\ &= ((h_1, k_1) \otimes (h_2, k_2)) \otimes (h_3, k_3).\end{aligned}$$

In a direct product, the two parts (the left and right sides of the ordered pair) act "independently" in that there is no "mixing" of the two halves. If we look at the set: $S = \{(h, e_k) : h \in \mathcal{H}\}$, it is easy to see that S is a group, and not only that, it is isomorphic to $\mathcal{H}$. Similarly, the set of all ordered pairs where the first element is e_h, the identity of $\mathcal{H}$, also forms a group that is isomorphic to $\mathcal{K}$.

If a group $\mathcal{G}$ can be represented as a direct product of two other groups, you are effectively factoring it in that same way that you factor integers. Once you have the factorization, you can learn almost everything about $\mathcal{G}$ by studying the properties of the two groups whose direct product $\mathcal{G}$ is.

There are many subgroups of $\mathcal{R}$ that are direct products (or to be precise, are *isomorphic* to direct products. In many cases, these arise in a natural way, meaning that the two groups making up the direct product really do "have nothing to do with each other" in the larger group.

Perhaps the simplest example is the group generated by the two moves F amd B. Clearly, twisting the front face has no effect on the back face and vice-versa. This group is like the direct product of the group generated by just F and the group generated by just B. Each of those groups has four elements, so the direct product will have 16. If we choose not to move the front face (in other words, we apply the identity to it), but allow any operation on the back face, that's like choosing (e_h, k), in our definition of the direct product above. It is clear that leaving the front face alone and twisting the back is by definition just like twisting the back. Similarly, if the back face is left unmoved, the for possible operations are equivalent to the group generated by F.

Are there any others? The answer is: Plenty! Consider the set of operations that leave all the corners fixed. We've seen a bunch of these in our cube-solving macros that flip or cycle edges. There's a similar set of operations that leave the edges fixed and move the corners. The direct product of these two groups is huge (but not the entire group $\mathcal{R}$).

There are many, many, smaller examples. Consider the group of operations that leave everything in place except for the four edge cubies on the bottom face and another group that leaves everything fixed except the four corner cubies on the right face. Although the bottom and right faces intersect, the two groups move no cubies in common, so the direct product of those two groups would be a group that jumbles the four bottom edge cubies and the four right corner cubies, and leaves all the rest of the cubies fixed.

There are thousands of other possibilities. Try to think of some of them yourself.

15 The Subgroup generated by U and R

In the previous section, we stated that the group generated by the two elements U and R has order 73483200—it's a relatively huge subgroup.

The proof that this is the case will be based upon a few macros that can be discovered by computer search: $P_1 =$ mRUrURUUrUU and $P_2 =$ urUURUrURURUUruRur. These can be combined to make the very interesting: $P_3 = P_1$UP_2u.

Macro P_2 cycles three edge cubies and macro P_3 rotates three corner cubies in place. Both leave everything else in place. Furthermore, P_1u moves some corners around, but leaves all the edge cubies in place except that it exchanges the UL and UF edgies. It's fairly clear that since this single exchange is possible, that, combined with conjugation, can produce any rearrangement we like of the edge cubies, assuming that we don't care what happens to the corner cubies in the process.

But in a sense the edge cubies can never be flipped. Imagine drawing an arrow on each of the seven edges so that the set of arrows around both the up- and right-faces form loops. (This would look like a figure-8 if it were flattened out.) Clearly, if all we can do is twist those two faces, all the arrows will point the same way after any sequence of such moves. Thus the orientation of every edgie is completely determined in every position for any combination of U and R moves.

So if we only care about the group of movements of the edge cubies where the locations and rotations of the corner cubies don't make any difference, we obtain the symmetric group on 7 elements, those being the 7 edge cubies on the two adjacent faces. The order of the edge group is $7! = 5040$.

It can be shown (but we will not do it here) that the group of permutations of the corner cubies, ignoring twisting, has order 120. Since any set of three of them can be twisted $1/3$ turn and there are 6 corner cubies, once the orientations of 5 of them are determined, the sixth is determined, but the orientations of the other 5 are arbitrary. Thus the size of the corner group (ignoring the movements of the edges) is $120 \times 3^5 = 29160$. If all possible edge configurations could occur with all corner configurations, the entire U-R subgroup would consist of $5040 \times 29160 = 146966400$ elements.

But since this is a subgroup of $\mathcal{R}$ we are still constrained by the fact that every permutation of all the cubies (edgies and cornies) must be even, so only half of the positions are possible, for an order of $146966400/2 = 73483200$. This is the size of the group, although we will not prove that, either.

16 Group Isomorphism

In Section 6.3 we mentioned the fact that the group of all permutations on three objects behaved exactly the same as the set of symmetries of an equilateral triangle. We said that two groups that behave identically are called isomorphic.

Two groups are called isomorphic if there is an isomorphism between them. An isomorphism is a $1-1$ onto mapping from one group to the other that preserves the group operation.

To state this in a formal way, suppose that we have two groups, $\mathcal{G}$ and $\mathcal{H}$ where $*$ is the group operation in $\mathcal{G}$ and $\odot$ is the group operation in $\mathcal{H}$. We say that $\mathcal{G}$ is isomorphic to $\mathcal{H}$ if there exists a $1-1$ onto function $f : \mathcal{G} \rightarrow \mathcal{H}$ satisfying the following condition: for every pair $g_1, g_2 \in \mathcal{G}$, if $f(g_1) = h_1$ and $f(g_2) = h_2$ then $f(g_1 * g_2) = h_1 \odot h_2$.

A "$1-1$ onto function" is one that matches up every element from one set with every element of another so that all the elements of both are used and each one maps into exactly one other one.

There may be more than one way to do the mapping, but that doesn't matter—as long as there is at least one way to do it, the two groups are said to be isomorphic.

If two groups are isomorphic, they are virtually identical—it's almost as if you just made a mistake and used different names for exactly the same things. Now this does not mean it's easy to find such isomorphisms or to prove that one exists, but it does mean that if you do find one, you have a way to translate the different names back and forth.

As an example, one very simple group is the set of permutations on the cube that can be achieved by just twisting the right face. This group obviously has four elements: 1, F, F^2 and F^3. Let's call this group $\mathcal{G}$. A different group, call it $\mathcal{H}$, is the set of all permutations that can be achieved by twisting the up face. Thus $\mathcal{H}$ consists of 1, U, U^2 and U^3. It is clear that these two groups behave identically, and we can show that they are isomorphic by checking that the function f that maps $1 \rightarrow 1$, $\mathsf{F} \rightarrow \mathsf{U}$, $\mathsf{F}^2 \rightarrow \mathsf{U}^2$ and $\mathsf{F}^3 \rightarrow \mathsf{U}^3$ satisfies the conditions to be an isomorphism.

Note that the f above is not the only function with the appropriate properties. Another function that is an isomorphism maps $1 \rightarrow 1$, $\mathsf{F} \rightarrow \mathsf{U}^3$, $\mathsf{F}^2 \rightarrow \mathsf{U}^2$ and $\mathsf{F}^3 \rightarrow \mathsf{U}$. This just shows that if you swap clockwise and counter-clockwise rotations, they behave pretty much the same way.

To say that two groups are isomorphic is to put quite a strong restriction on their relationship. If we simply require that the function f preserve the group operation, but do not require that it be $1-1$ or onto, we have a different relationship that is called a homomorphism. Of course an isomorphism is a special case of a homomorphism, but there are many, many more homomorphisms available.

Isomorphisms are usually used to show that two apparently different groups are essentially identical, so they are usually constructed to relate different groups. They can, of course, be mappings of a group into itself (in which case they are often called automorphisms). Since the most important group from the point of view of this paper is the cube group $\mathcal{R}$, many of the examples that follow will be homomorphisms from that group into itself. In general, a homomorphism can relate any group to any other, but in the special case where a homomorphism maps a group into itself, it can be called an endomorphism.

Here are a couple of examples of automorphisms involving $\mathcal{R}$ and subgroups of it. First, consider the group $\mathcal{G} = \{\ \mathsf{1}, \mathsf{F}, \mathsf{F}^2, \mathsf{F}^3\ \}$. There are two automorphisms of $\mathcal{G}$. The first is not too interesting, since it maps every element to itself, but another automorphism takes $\mathsf{1}$ and F^2 to themselves but exchanges F and F^3. From the point of view of the cube, these two groups are mirror images—the counter-clockwise moves are swapped with the clockwise groups, but other than that, their behavior is the same.

There are plenty of automorphisms of the full group $\mathcal{R}$ onto itself. Imagine two identical physical cubes originally in the same orientation, and we take the one on the right and rotate it 90° clockwise about its front face. At this point, the front and back faces have the same orientation, but the top, right, bottom, and left sides of one correspond to the right, bottom, left and top sides, respectively, of the other. If we consider any move sequence on one and replace the letters in it as follows:

F⟶F	B⟶B	U⟶R	R⟶D	D⟶L	L⟶U
f⟶f	b⟶b	u⟶r	r⟶d	d⟶l	l⟶u

then the moves on the two cubes will behave in exactly the same way. There are (counting reflections) 48 automorphisms that are very similar to this and correspond to the symmetries of a cube.

We will not prove this here, but the Rubik cube group $\mathcal{R}$ is so large and complicated that it contains subgroups that are isomorphic to every existing group of order less than 13, and it also contains every abelian group of order less than 26. (Remember that an abelian group is a group where the operation is commutative.)

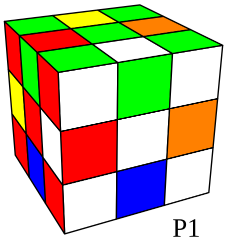
For a challenge, you might try to find some of those subgroups. For example, consider the quaternion group. This is a group of order 8 that consists of $\{1, i, j, k, -1, -i, -j, -k\}$ that satisfies the relations: $i^2 = j^2 = k^2 = ijk = -1$.

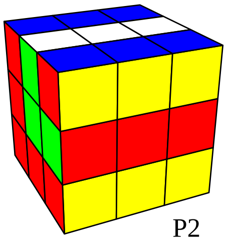
17 Pretty Patterns

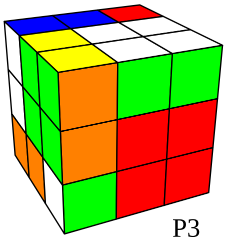
In addition to just solving the cube, it is possible to create many pretty patterns. You can use the **Rubik** program to search for your own pretty patterns. Use the “Input Cube” command to draw in whatever pattern you want, and if it is a legal pattern, **Rubik** can find a sequence of moves to “solve” it. The inverse of that solution will generate the pattern from a solved cube.

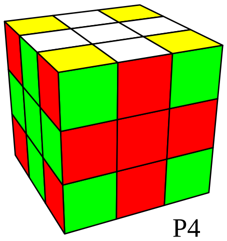
In the list below, to arrive at each of the patterns you need to begin with a solved cube and apply the given macro.

These patterns were found on the internet—obviously, the names are not universal. The itemized list below contains the move sequence to reach the pattern from solved, together with the name of the macro.

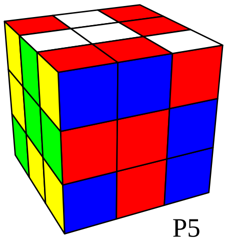

P1

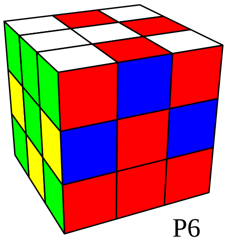

P2

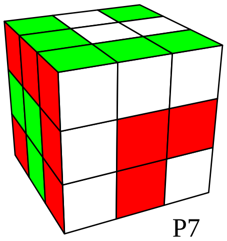

P3

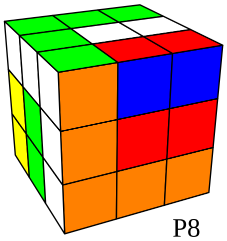

P4

- 2(4(*RU)>R>D)4(*RU): Superflip. Figure P1.


P5


P6


P7


P8

- RDRFrfBDrubUDD: Green Mamba. Figure P2.
- FFDFFDDLLULLuLLBDDRR: Six Square Cuboids. Figure P3.
- uFFUUlRFFUUFFLru: Christmas Cross. Figure P4.
- RRDuLdLLRBRRUBUrfdFu: Twisted Duck Feet. Figure P5.
- UULLrbRRurDRFFLrFDLLUU: Plummer’s Cs. Figure P6.
- LBBDRbFdlRdUfRRu: Anaconda. Figure P7.
- dFdLBDDFFURbURRFdRFUU: Striped Cube. Figure P8.

18 Miscellaneous Short Topics

The topics here are interesting, but are a bit too short to deserve to have an entire section devoted to them. They are not in any particular order.

18.1 Rotation of the Center Cubies

On a standard cube, it is impossible to tell after a series of twists whether the center cubie has the same orientation. Some cubes made for advertising purposes have images on some of the faces, and if you simply apply the standard cube solution methods, you’ll find that you’ve got everything correct except that some of the center faces are rotated from their solved positions. You can observe this with a physical cube: Put a sticker on the U face of a solved cube with an arrow pointing toward the front face and similarly one on the front face pointing toward the up face. Then apply the following move:

$$\mathsf{URLUUrlURLUUrl} = 2(\mathsf{URLUUrl})$$

At the end, you will find that the arrow on the U face has rotated by 180° (and if you had similarly marked the orientation of the other faces, you would find that this transformation leaves them unchanged. Here is a transformation that twists U and D by 90° each in opposite directions:

$$\mathsf{RlFFBBRlURlFFBBRld}$$

Although at first it doesn't look like it, the macro above is (as you might expect) a commutator. Let $\mathcal{P} = >$F$>$FRlFFBBRl and let $\mathcal{Q} =$ U. Then the macro above is $\mathcal{P}\mathcal{Q}\mathcal{P}^{-1}\mathcal{Q}^{-1}$.

And finally, here is one that rotates the U center by 90° clockwise and at the same time rotates the F center cubie 90° counter-clockwise:

FbLrUdfDuRlBfU

This is also a commutator. Let $\mathcal{P} =$ FbLrUd$>$R and $\mathcal{Q} =$ u, and the macro is equivalent to $\mathcal{P}\mathcal{Q}\mathcal{P}^{-1}\mathcal{Q}^{-1}$.

In exactly the same way as we have seen before, there is a sort of parity associated with the total twist of the center cubies: the grand total of the twists must add to an even multiple of 180°.

18.2 Superflip

In Section 11 we mentioned the permutation called superflip that flips every edge cubie in place and leaves all the corner cubies unchanged. It has been shown that there is a 24 move macro that achieves superflip and it has also been shown that superflip *requires* 24 quarter-turns so that it serves to prove that 24 is a solid lower bound on God's number (the maximum length of the best possible solutions to any jumbled cube).

The superflip permutation is interesting in that it is the *only* permutation that commutes with every element of $\mathcal{R}$. The set of all permutations that commute with every element in the group is called the center of the group, and the center of a group is, in fact, a subset of the group. The center of $\mathcal{R}$ is a two-element group consisting of the identity and superflip.

Since at first glance, a solved cube with superflip applied looks pretty messed up, you can hand a superflipped cube to a friend, have him make two or three moves and hand it back to you while your eyes are closed. You then apply superflip, and then say something like "...and now it's almost done...", open your eyes, and undo the final few twists.

18.3 The Whole-Cube Group

One interesting group that is very easy to study using your physical cube is the whole-cube group—the rigid symmetries of the cube. It is pretty obvious that there are 24 such symmetries since the top face can be moved to any of 6 faces, and once there, can be rotated into any of four positions for a total of $6 \times 4 = 24$ symmetries. (Notice that if we also allowed mirror images, there would be 48 total symmetries.)

To work with the group, we'll need some names for the group elements. In this section only, we'll use the name F to represent a rotation of the whole cube clockwise by 90° instead of the "$>$F" used by **Rubik** and similarly for the other rotations.

If you wish to experiment with these permutations, it's probably a good idea to label the face cubies on your physical cube with "U", "D", et cetera. The advantage of working with this group is that there's no way to scramble your cube accidentally. You might try to find subgroups of this group, for example, and to see what those subgroups amount to geometrically.

It's a bit of a mess to describe all 24 elements. There are a few obvious ones: 1 (the identity), F, L, F, R, B, D, FF, RR and UU. (We don't need LL, BB and DD since LL $=$ RR, et cetera.) The other 14 permutations can also be expressed as rotations, but unfortunately, about oddball axes. If we consider the four axes that connect opposite corners of the cube (like corner URF with corner DLB, et cetera) there are two rotations ($120°$ and $240°$) that map the cube to itself for 8 more permutations. We also have the 6 axes connecting the centers of opposite edges of the cube, and a rotation of $180°$ about each of these is also a rigid symmetry of the cube.

But rather than invent new names for these rotations, we'll just list the other ones as products of the primitive face rotations that we already have. For example, the rotation of $180°$ about the axis passing through the centers of the edge cubies UL and DR is LFU.

The first table below shows the definitions of the moves in an easier to read form: a cube that has been opened up. The entry labeled "1" shows the initial configuration and the others show how those faces are rearranged by the various rotations. This is very useful since the three-move combinations chosen to represent the last six permutations are somewhat arbitrary: BLU $=$ DRF, for example.

The second is the multiplication table for the group of rigid moves of the cube. In the list below, "F" means to grab the front face and turn the entire cube clockwise by a quarter-turn, et cetera. To multiply RR with FL, for example, choose the entry in the column with RR on top and the row with FL on the left. The product is the permutation in that column and row: BR.

This group contains 24 elements, and is isomorphic to the symmetric group on 4 objects (the group of all permutations of four objects. To see why this is, notice that a cube has four diagonals and that with an appropriate twisting in space, those four diagonals can be mapped in themselves in all possible ways. Since the entire group is effectively the symmetric group on four objects, you can find the alternating group on four objects as a subset. Can you figure out which elements are members of the alternating group?

1: U L F R B D	F: L D F U B R	R: F L D R U B	U: U F R B L D	B: R U F D B L	L: B L U R D F
D: U B L F R D	FF: D R F L B U	RR: D L B R F U	UU: U R B L F D	FR: F D R U L B	RB: R F D B U L
BL: B U R D L F	LF: L F U B D R	LB: R B U F D L	FL: B D L U R F	RF: L B D F U R	BR: F U L D R B
FRU: F R U L D B	RUF: D B R F L U	LBD: R D B U F L	BUR: D F L B R U	LFU: L U B D F R	BLU: B R D L U F

Table 4: Whole Cube Move Definitions

	1	F	R	U	B	L	D	FF	RR	UU	FR	RB	BL	LF	LB	FL	RF	BR	FRU	RUF	LBD	BUR	LFU	BLU
1	1	F	R	U	B	L	D	FF	RR	UU	FR	RB	BL	LF	LB	FL	RF	BR	FRU	RUF	LBD	BUR	LFU	BLU
F	F	FF	RF	FR	1	LF	FL	B	LFU	LBD	RUF	R	U	FRU	L	BUR	BLU	D	LB	BL	RR	BR	UU	RB
R	R	FR	RR	RB	BR	1	RF	FRU	L	BLU	LBD	BUR	B	U	D	F	RUF	LFU	UU	LB	FL	LF	BL	FF
U	U	LF	FR	UU	RB	BL	1	BUR	RUF	D	FRU	LBD	BLU	LFU	B	L	F	R	BR	FF	LB	RR	RF	FL
B	B	1	RB	BL	FF	LB	BR	F	LBD	LFU	U	BLU	RUF	L	FRU	D	R	BUR	LF	FR	UU	FL	RR	RF
L	L	FL	1	LF	BL	RR	LB	BLU	R	FRU	F	U	LFU	BUR	RUF	LBD	D	B	FF	RF	FR	RB	BR	UU
D	D	RF	BR	1	LB	FL	UU	RUF	BUR	U	R	B	L	F	LBD	BLU	LFU	FRU	FR	RR	RB	FF	LF	BL
FF	FF	B	BLU	RUF	F	FRU	BUR	1	UU	RR	BL	RF	FR	LB	LF	BR	RB	FL	L	U	LFU	D	LBD	R
RR	RR	LBD	L	BUR	LFU	R	RUF	UU	1	FF	FL	LF	BR	RB	RF	FR	LB	BL	BLU	D	F	U	B	FRU
UU	UU	LFU	FRU	D	LBD	BLU	U	RR	FF	1	BR	LB	FL	RF	RB	BL	LF	FR	R	BUR	B	RUF	F	L
FR	FR	FRU	RUF	LBD	R	U	F	BR	BL	FL	LB	RR	RB	UU	1	LF	FF	RF	D	B	L	LFU	BLU	BUR
RB	RB	U	LBD	BLU	BUR	B	R	LF	LB	RF	UU	FL	FF	BL	BR	1	FR	RR	LFU	FRU	D	L	RUF	F
BL	BL	L	U	LFU	BLU	RUF	B	FL	FR	BR	LF	UU	RF	RR	FF	LB	1	RB	BUR	F	FRU	LBD	R	D
LF	LF	BUR	F	FRU	U	LFU	L	RB	RF	LB	FF	FR	UU	BR	BL	RR	FL	1	B	BLU	RUF	R	D	LBD
LB	LB	D	B	L	RUF	LBD	FRU	RF	RB	LF	1	BL	RR	FL	FR	UU	BR	FF	F	R	U	BLU	BUR	LFU
FL	FL	BLU	D	F	L	BUR	LBD	BL	BR	FR	RF	1	LF	FF	RR	RB	UU	LB	RUF	LFU	R	B	FRU	U
RF	RF	RUF	LFU	R	D	F	BLU	LB	LF	RB	RR	BR	1	FR	FL	FF	BL	UU	LBD	L	BUR	FRU	U	B
BR	BR	R	BUR	B	FRU	D	LFU	FR	FL	BL	RB	FF	LB	1	UU	RF	RR	LF	U	LBD	BLU	F	L	RUF
FRU	FRU	BR	FF	LB	FR	UU	LF	R	BLU	L	B	RUF	LBD	D	U	LFU	BUR	F	1	RB	BL	RF	FL	RR
RUF	RUF	LB	BL	RR	RF	FR	FF	D	U	BUR	L	LFU	R	LBD	F	FRU	B	BLU	FL	1	LF	UU	RB	BR
LBD	LBD	UU	LB	FL	RR	RB	FR	LFU	B	F	D	L	BUR	BLU	R	U	FRU	RUF	RF	BR	1	BL	FF	LF
BUR	BUR	RB	FL	FF	LF	BR	RR	U	D	RUF	BLU	F	FRU	B	LFU	R	LBD	L	BL	UU	RF	1	LB	FR
LFU	LFU	RR	LF	BR	UU	RF	BL	LBD	F	B	BUR	FRU	D	R	BLU	RUF	L	U	RB	FL	FF	FR	1	LB
BLU	BLU	BL	UU	RF	FL	FF	RB	L	FRU	R	LFU	D	F	RUF	BUR	B	U	LBD	RR	LF	BR	LB	FR	1

Table 5: Whole Cube Multiplication Table

A Unjumbling the Cube

The most obvious way to unjumble a cube is to pop it apart with a screwdriver and then reassemble it in the “solved” state. If you have the **Rubik** program, however, there is an easier way.

A.1 The Screwdriver Method

First, the screwdriver approach: Turn one face of your cube by 45 degrees. Next, insert the tip of the screwdriver under one of the edge cubies on that face that you just turned and pry it up. The edge cubie will pop out. Every other cubie can now be easily removed by hand, but pay attention to the first few you remove so you’ll remember how to fit them back together.

To restore the cube, notice that every cubie is different, and that there are two types—corner cubies with three colored facelets and edge cubies with two colored facelets. Notice that if you know those three or two colors, there is only one place in the final cube where the cubie can go, relative to the six face cubies that are all connected in the central “skeleton”.

After the cube is disassembled completely, put it together, cube by cube, where each cube is placed in its correct position relative to the central skeleton. Save an edge cubie (one with two colors) for last, and to insert it, turn the face with the missing cubie 45 degrees relative to the rest of the cube, hook one corner of the cubie connector into the almost-reconstructed cube, and push it in until it snaps into place.

A.2 The **Rubik** Method

The **Rubik** program has a built-in solver that can unjumble any cube. What you will do is to enter the cubie configuration of your currently-jumbled cube and then ask **Rubik** to solve it. The solution is simply a list of sides to twist that will bring the jumbled cube to solved.

Before going to the trouble of entering your own cube, first see how the solving feature works. Fire up **Rubik** and then click on the Jumble Cube button that you will find in approximately the center of the control area to the right of the drawing of the cube. This will jumble the cube as if you had randomly turned hundreds of faces in random directions.

Next, click on the Solve Cube button that lies just to the right of the Jumble Cube button. This will take a while, depending on how fast your computer is. On very slow older machines it may take up to a couple of minutes. In fact the first time you solve a cube after starting up **Rubik**, it takes even longer because there is some initialization required of the solver that only has to be done once. On a relatively quick machine (as of 2003), it takes, on average, about 15 seconds to initialize and solve. But beware! Sometimes, for particular cubes it takes a lot longer. An example that seems to take a long time for some reason is 6(FFRRUU).

When **Rubik** has a solution, a little window will pop up telling you that it’s done, and an encoded solution will appear in the window labeled “Current Macro”. The solution consists of a series of letters and each letter corresponds to one quarter-twist of a face. The letters “U”, “L”, “F”, “R”, “B” and “D” (and the lower-case versions of those same letters) stand for “up”, “left”, “front”, “right”, “back” and “down”, respectively.

If the letter is in upper-case, it means that you should grasp that face with your right hand and turn it a quarter-turn in the direction pointed to by your right thumb. If the letter is in lower-case, turn it a quarter-turn in the other direction. Do *not* lose the orientation of your cube—keep whatever face was initially up pointing up and whatever face was in front should remain facing front.

Click on the OK button in the little window to make it go away. Now, you can single step through the solution by clicking repeatedly on the **right-arrow** key.

When you have entered your own cube’s configuration and get a solution, the best way to solve your cube if you’re not totally familiar with the move descriptions is to click on the arrow key once, then find a face on your cube that you can twist to make it look exactly like the cube on the screen. Then press the **arrow-key** again, do one more step, and so on. **Rubik**’s solutions are typically less than 30 twists long, so it will not take long once you have entered your cube’s colors into the program. (The little cube visible in the upper-right of the drawing area shows what the back of your cube should look like, which can be quite helpful.)

If you are making these moves on your physical cube and you suddenly notice that you’re mixed up, it is probably easiest just to re-enter your cube (you won’t need to change much) and to click on the Solve Cube button again. The new solution will probably be much shorter since you will presumably have made some progress toward the solution before you committed your error.

Note: If your cube’s colors are different from those in Rubik you can change Rubik’s colors to match. See the documentation that comes with the program.

So all you need to know how to do is to load your cube’s color pattern into **Rubik**. To do so, click on the Input Cube entry in the Edit pull-down menu. A new window will appear that displays and unwrapped version of the cube. One of the cubies is highlighted. To set its color, simply click on the appropriate color from the palette of colors at the bottom of the window. After each color is entered, the highlighted cubie advances. If you make a mistake, simply click on the cubie that’s in error and click on its color, and so on.

When you have your colors correct, click on the Finish button and the results will be displayed in the window. If you made a mistake, use the Input Cube button again, and fix the few bad cubie colors. **Rubik** can check for some errors in your input, but not all. If it does report an error, you have certainly done something wrong, so you’ll need to use the Input Cube command again.

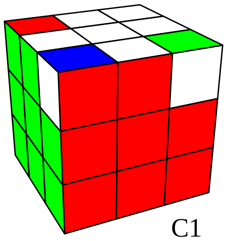
If you’ve never done this before, you may have to repeat it a couple of times before you succeed in solving your cube.

B Cube-Solving Macros

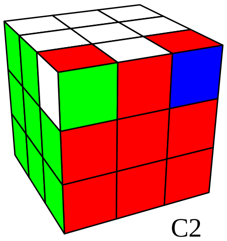
Warning: If you enjoy working out puzzles for yourself, don’t read this section! It contains a detailed list of macros that are useful for solving the cube. It’s much more fun if you work out your own set and only then take a look at the collection found here.

The macros in the list below do interesting things to the cube. There is a short description of the effects of each one, but to see exactly what each does, run **Rubik** and apply each one to a solved cube. For example, the first macro below "cycles three corners". Which three corners? Test the macro in **Rubik** to see. Notice that there are duplications with respect to what the macro does. This is because in most cases one of the versions, although longer, is much easier to memorize. Macros that are commutators, for example, tend to be easier (at least for me) to memorize.

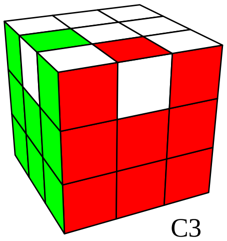
You'll see that the list below includes the set of macros that are built-in to **Rubik**. This first set is very rigid in that the macros here do very specific minimal changes to the cube. These are the sorts of macros that you would use when you are very near a solution. The number in parentheses after each macro is the number of quarter-turns required to perform it. All of the macros below were performed on a cube with yellow on the bottom, orange on the back, and blue on the right.



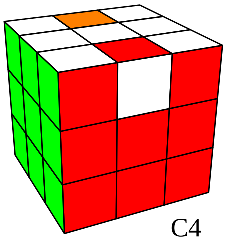
C1



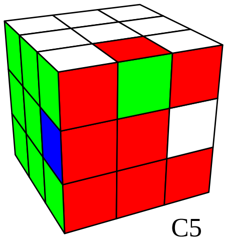
C2



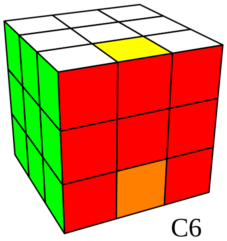
C3



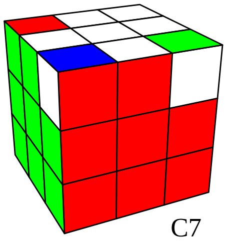
C4



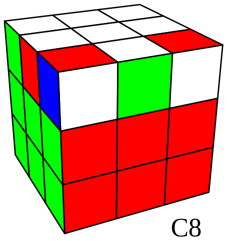
C5



C6



C7



C8

- fUBuFUbu (8Q): A commutator that cycles three corners and leaves the rest of the cubies intact. If $\mathcal{P} =$ UBu, this macro is f$\mathcal{P}$F$\mathcal{P}^{-1}$. Figure C1.
- LdlfdFUfDFLDlu (14q): Rotates two corner cubies in place and does not move any of the other cubies. This is also a commutator. If $\mathcal{P} =$ LdlfdF, then the macro is $\mathcal{P}$U$\mathcal{P}^{-1}$u. Figure C2.
- FRBLUlUbrfluLu (14Q): Flips two adjacent edge cubies. It is not quite a commutator: Let $\mathcal{P} =$ FRB and let $\mathcal{Q} =$ UlU. Then this macro is $\mathcal{P}$L$\mathcal{Q}\mathcal{P}^{-1}l\mathcal{Q}^{-1}$. It's actually sort of like a triple commutator since the inverse of L is l. Figure C3.
- RlFFLrdRlFLruRlfLrDRlffLrU: A pure commutator to flip two adjacent edge cubies. If $\mathcal{P} =$ RlFFLrdRlFLr, the macro is $\mathcal{P}$u$\mathcal{P}^{-1}$U. Figure C3.

- LfUlFbUrFuRfBu (14Q): Flips two opposite edge cubies. This one is just short, and has little else to recommend it. Figure C4.

- *LU*LU*LUU*RU*RU*RUU (20Q): Flips two opposite edge cubies in place. This is easy to remember and seems faster than the 20 quarter-turns that it requires since six of the moves are slice moves. This is a commutator: if $\mathcal{P}$ =*LU*LU*L, then the macro is: $\mathcal{P}$UU$\mathcal{P}^{-1}$UU. (Remember that UU is its own inverse.) Figure C4.

- UFFurdlFFLDR (12Q): Cycles three edge cubies, but is not a commutator. Figure C5.

- RlUUrLFF (8Q): Cycles three slice edge cubies. Very fast and can be thought of as being a commutator when viewed as being composed of two slice moves: *LFF*RFF, since *L and *R are inverses. Figure C6. (The third cubie in the cycle is the UB cubie.)

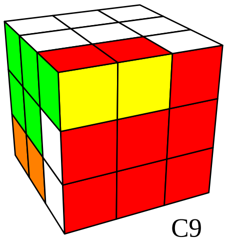
- fUBuFUbu (8Q): Cycles three corner cubies. It's also a commutator: if we let $\mathcal{P} =$ Ubu, the macro is f$\mathcal{P}$F$\mathcal{P}^{-1}$. Figure C7.

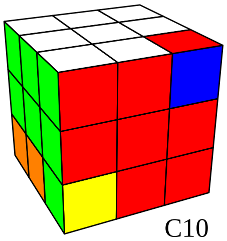
- rURurUFRbRBRfRR (15Q): Swaps two corners and two edges, and does some flipping and rotating of those cubies as well. It leaves the rest of the cubies unchanged. Figure C8.

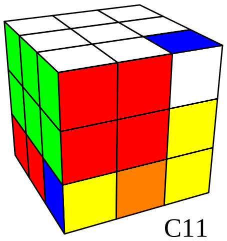
The rest of the macros in this section are used to solve the cube early in the solution process. They are generally quite fast, but they trash varying amounts of the rest of the cube. The ones you choose to use depend on your overall cube-solving strategy. For example, if you start by getting all the cubies on one face correct, you will usually do that either by getting all the corners followed by all the edges or vice-versa. If you do corners first, then the moves to place the edges must preserve the corners; if you place the edges first, you don't care what the edge-setting moves do to the corners and so on.

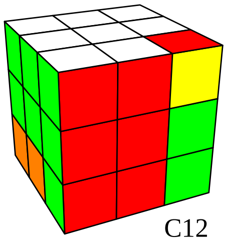
It's easiest to see what each one does by applying them to a solved cube with rubik.

If you decide to solve the top face by doing the edges first, then the corners, here are some macros to do the top face. All of the macros below were performed on a cube with yellow on the bottom, orange on the back, and blue on the right.

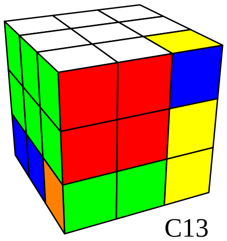

C9

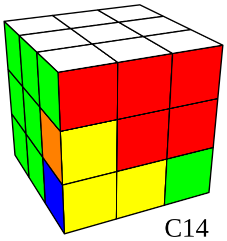

C10

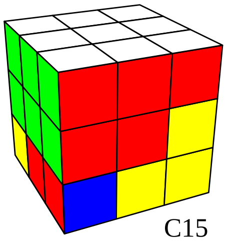

C11

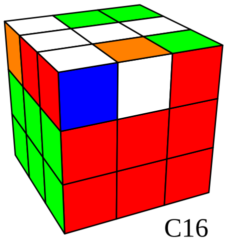

C12

- DRfr (4Q): Moves an edge cubie from the bottom front face to the top front face and flips it relative to what FF would do. This move is for getting the top face correct. It leaves all the other cubies on the top where they were, but it does twist one of the corner cubies in place. Notice that if the edge cubie is in the correct position on the top face but is flipped, you can do an FF followed by this macro to flip it. Figure C9.


C13


C14


C15


C16

- FDfdFDf (7Q): Rotates the UFL corner cubie clockwise in place. This leaves all the other cubies on the top face exactly as they were. Figure C10.

- rdRDrdR (7Q): Same as above, but rotates the corner cube counter-clockwise. Figure C11.

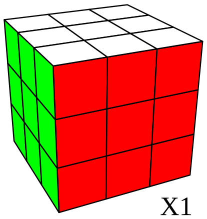
- FDf, rdR (3Q): Brings a corner cubie from the bottom to the top face directly above it and rotates it counter-clockwise or clockwise on the way up. No other cubies on the top face are altered. Figure C12 (for FDf).

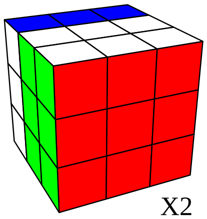
- rDDRDrdR (8Q): Brings a corner cubie up from the bottom to the top face directly above it, and gives it a 180° flip on the way up. This macro has no effect on any of the other cubies in the top face. Figure C13.

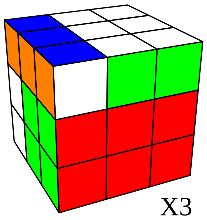
- fdFDLDl, FDfdrdR (7Q): Moves an edge cubie from the lower face to the middle face without altering the top face at all. Figure C14 (for fdFDLDl) and C15 (for FDfdrdR).

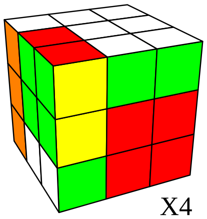
- BULulb (6Q): Cycles three edge cubies on the top face. This mixes up the top-face corner cubies but has no effect on the lower two levels. Figure C16.

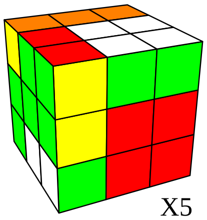
C Make Cover Cube

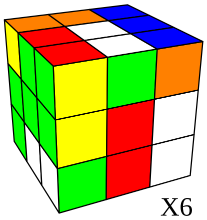

X1

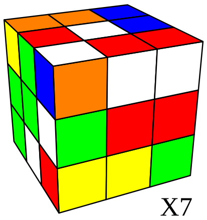

X2

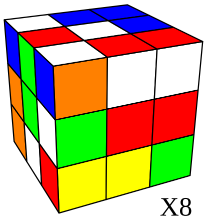

X3

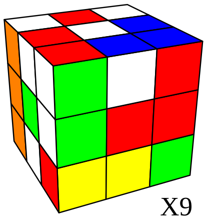

X4

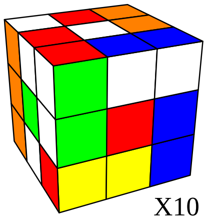

X5

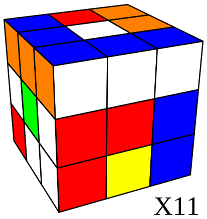

X6

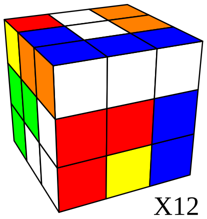

X7

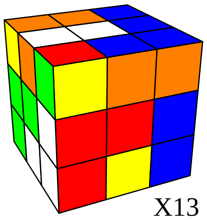

X8

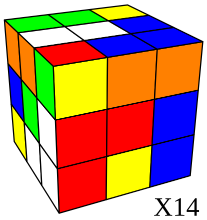

X9

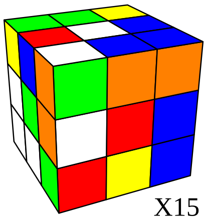

X10

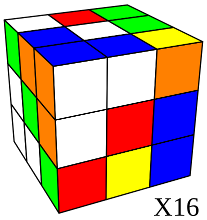

X11

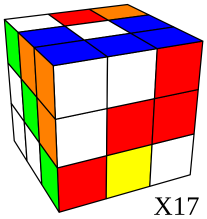

X12

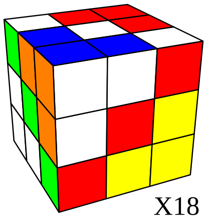

X13


X14


X15


X16


X17


X18

Here is a complete execution of the 17-move sequence BulbrfBUrLbubLURR that makes the picture on the cover of this article. Each step is exactly one quarter-turn.

Index