

A CHILD'S GARDEN OF GROUPS

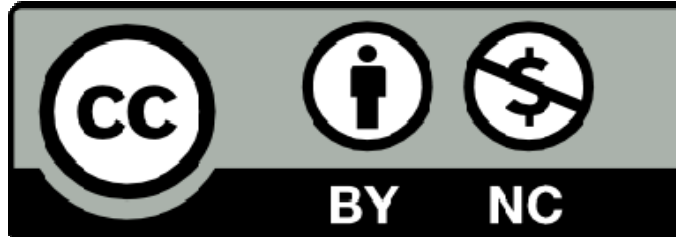
Subgroups, Cosets and Equivalence Relations!

(Part 2)



by

Doc Benton



Creative Commons License

You are free to:

- Share this work
- Adapt this work
- Attribute all original materials to Doc Benton

You are not free to:

- Charge money for this work; Knowledge is free!

CONTENTS (PART 2)

Introduction (Part 2)	1
Subgroups	2
Right Cosets	6
Left Cosets.....	12
Quotient Groups.....	14
Equivalence Relations	17
Something from Nothing Creation.....	21
Special Subgroups.....	23
Exploring Rubik's Cube with GAP.....	28
How to Use GAP (Part 2).....	33
Summary (Part 2).....	46
Practice (Part 2).....	47
Practice (Part 2) – Answers	50

INTRODUCTION (PART 2)

In part 1 of this book, we introduced the concept of a mathematical *group*, *cyclic groups*, the connection between *groups*, *permutations*, and *symmetry*, some of the basics of GAP software, *group actions*, and the relationship between *group theory* and Rubik's cube. In this second part we'll build upon the introduction to *group theory* given in part 1. In particular, we'll introduce both the concept of a *subgroup* and that of a *quotient* or *factor group*. At the same time, we'll also introduce additional commands for the software program GAP, and we'll use GAP to do some additional analysis of Rubik's cube. Everything builds upon everything else, so it's important to have first mastered the lessons in part 1. Nonetheless, even though several more parts will follow, if you make it through part 2, then you will definitely be in the top 1% as far a *group theory* knowledge goes! Enjoy!

SUBGROUPS

Recall that the *Rubik's cube group* contains 43,252,003,274,489,856,000 distinct elements and that these elements correspond to the number of distinct permutations or "scramblings" that are possible of the colored facelets on Rubik's cube. Recall also that all the elements of this *group* can be generated by the moves R , L , U , D , F , and B where each of these moves corresponds, respectively, to a clockwise quarter-turn of the right, left, up, down, front, or back face of the cube.

Now, however, let's consider just those elements of the *Rubik's cube group* that can be generated by rotating only the right face a quarter-turn clockwise (R). Multiples of this sort will generate a *cyclic group* of four elements where the distinct elements can be denoted by $\{(), R, R^2, R^3\}$. The primary thing to learn from this example, though, is that it is possible for a smaller *group* to be contained within a larger *group* just as our newly found four element *group* $\{(), R, R^2, R^3\}$ is a part of the complete *Rubik's cube group* of 43,252,003,274,489,856,000 elements. Whenever this happens, we say that the smaller *group* is a *subgroup* of the larger *group*. Furthermore, if we denote the smaller *group* by H and the larger *group* by G , then we write $H \leq G$ to indicate that H is a *subgroup* of G . Also, if we want to emphasize that H is a *subgroup* of G that is not equal to all of G , then we write $H < G$ instead, and we say that H is a *proper subgroup* of G .

Every *group* G has at least two *subgroups*, itself and the *subgroup* consisting of the identity element, G and $\{() = \{e\}$. Also, if we take any element a of a *group* G and if we start looking at all the distinct *integer* powers of that element, $\{\dots, a^{-2}, a^{-1}, a^0 = () = e, a^1 = a, a^2, \dots\}$, then those powers will form either a *finite* or an *infinite cyclic group* that will also be a *subgroup* of G . Furthermore, the *subgroup*

generated by a single element a is frequently written as $\langle a \rangle$, and a *subgroup* generated by two elements a and b is denoted by $\langle a, b \rangle$. Likewise, a *group* or *subgroup* generated by three elements a , b , and c is denoted by $\langle a, b, c \rangle$.

Now let's take a look at the following multiplication table for a *permutation group* G that can be generated by two elements, $(1,2)$ and $(1,3)$.

```
gap> a:=(1,2);
(1,2)
gap> b:=(1,3);
(1,3)
gap> g:=Group(a,b);
Group([ (1,2), (1,3) ])
gap> Size(g);
6
gap> Elements(g);
[ (), (2,3), (1,2), (1,2,3), (1,3,2), (1,3) ]
gap> ShowMultiplicationTable(g);
*
  | () (2,3) (1,2) (1,2,3) (1,3,2) (1,3)
  +-----+
() | () (2,3) (1,2) (1,2,3) (1,3,2) (1,3)
(2,3) | (2,3) () (1,2,3) (1,2) (1,3) (1,3,2)
(1,2) | (1,2) (1,3,2) () (1,3) (2,3) (1,2,3)
(1,2,3) | (1,2,3) (1,3) (2,3) (1,3,2) () (1,2)
(1,3,2) | (1,3,2) (1,2) (1,3) () (1,2,3) (2,3)
(1,3) | (1,3) (1,2,3) (1,3,2) (2,3) (1,2) ()
```

Using the notation that we just introduced for a *group* that is generated by one or more elements, we could write that $\langle (1,2),(1,3) \rangle = G$. Additionally, each particular element of this *group* G will generate a *cyclic subgroup* of G . Below are the results which we may obtain either through direct multiplication of permutations from left to right, or from using the multiplication table to do our calculations.

$$\begin{aligned}
\langle () \rangle &= \{ () \} = \{ e \} && \text{one element} \\
\langle (2,3) \rangle &= \{ (), (2,3) \} && \text{two elements} \\
\langle (1,2) \rangle &= \{ (), (1,2) \} && \text{two elements} \\
\langle (1,2,3) \rangle &= \{ (1,2,3), (1,3,2), () \} && \text{three elements} \\
\langle (1,3,2) \rangle &= \{ (1,3,2), (1,2,3), () \} && \text{three elements} \\
\langle (1,3) \rangle &= \{ (1,3), () \} && \text{two elements}
\end{aligned}$$

Notice that we are able to generate one *subgroup* of order 1, three distinct *subgroups* of order 2, and only one distinct *subgroup* of order 3 since the cycles (1,2,3) and (1,3,2) generate the same *group*. Also notice that no single element generates the entire group G , and, thus, G is not itself a *cyclic group*. Nonetheless, each element of the *group* generates a *cycle*, and the interaction between the elements in the *cycle* generated by (1,2) and the *cycle* generated by (1,3) is enough to generate the entire group. This is why I like to think of *cycles* as the building blocks for all *groups* just as the *Rubik's cube group* is generated by the *cycles* produced by R , L , U , D , F , and B . Furthermore, on Rubik's cube the right and left faces operate independently of one another. In other words, if you first rotate the right face a quarter-turn clockwise followed by a similar quarter-turn clockwise of the left face, then the result is the same as if you had first turned the left face and then the right face. Turns, respectively, of the right and left faces of Rubik's cube commute with one another! On the other hand, there is an interaction between turns of the right face and turns of the up face, and by that I mean that if you do a quarter-turn clockwise of the right face of Rubik's cube followed by a quarter-turn clockwise of the up face, then the resulting pattern on the cube is different from what would result if you had done the turn of the up face first. These turns do not commute with one another, and order makes a difference. The bottom line is that the moves R and L generate a *subgroup*, $\langle R, L \rangle$, of the *Rubik's cube group* that is *abelian* while the moves R and U generate a *subgroup*, $\langle R, U \rangle$, that is *nonabelian*.



$RL = LR$



RU



UR

Recall now that a *group* is a nonempty set with a *binary operation* that possesses four algebraic properties – *closure*, *associativity*, *existence of an identity*, and *existence of inverses*. If we have a nonempty subset of a *group* G such that all four properties are present under the *group multiplication*, then our subset forms a *subgroup*. However, if we want to prove that some nonempty subset is a *subgroup*, then we don't need to worry about *associativity* because we get that one for free. Since our nonempty subset lies within a *group*, we automatically know that the *associative law* is going to hold under the given *group* multiplication. And in fact, we only need to verify that two properties are present – *closure* and *existence of inverses*. For example, suppose H is a nonempty subset of a *group* G and that the elements of satisfy both the *closure property* and the *existence of inverses property*. Then if $a \in H$ (a is an element of H), it follows that $a^{-1} \in H$, too. Hence, given the *closure property*, we can now conclude that $a \cdot a^{-1} = e \in H$. Thus, H contains the *identity element*, and since we already know that the *associative law* holds, it follows that H is a *subgroup* of G , $H \leq G$.

RIGHT COSETS

In our answers to “Practice (Part 1)” we gave the multiplication table for the following *group* generated by the cycles (1,2) and (1,3).

```
gap> a:=(1,2);
(1,2)

gap> b:=(1,3);
(1,3)

gap> g:=Group(a,b);
Group([ (1,2), (1,3) ])

gap> Size(g);
6

gap> Elements(g);
[ (), (2,3), (1,2), (1,2,3), (1,3,2), (1,3) ]

gap> ShowMultiplicationTable(g);
*
-----
|  ()  (2,3) (1,2) (1,2,3) (1,3,2) (1,3)
-----
() |  ()  (2,3) (1,2) (1,2,3) (1,3,2) (1,3)
(2,3) | (2,3) () (1,2,3) (1,2) (1,3) (1,3,2)
(1,2) | (1,2) (1,3,2) () (1,3) (2,3) (1,2,3)
(1,2,3) | (1,2,3) (1,3) (2,3) (1,3,2) () (1,2)
(1,3,2) | (1,3,2) (1,2) (1,3) () (1,2,3) (2,3)
(1,3) | (1,3) (1,2,3) (1,3,2) (2,3) (1,2) ()
```

This *group* G that is generated would usually be written horizontally as $G = \{(), (2,3), (1,2), (1,2,3), (1,3,2), (1,3)\}$, but for this topic, I find it better to write it vertically as follows:

$$G = \left\{ \begin{array}{c} () \\ (2,3) \\ (1,2) \\ (1,2,3) \\ (1,3,2) \\ (1,3) \end{array} \right\}$$

Also, let's consider the *subgroup* H of G that is generated by the cycle (1,2,3). The elements of H are:

$$H = \left\{ \begin{array}{c} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\}$$

And now we're going to give a very important definition, that of a *right coset*.

Definition: If G is a *group* and if H is a *subgroup* of G and if $g \in G$, then $Hg = \{h \cdot g \mid h \in H\}$. In this case, we call Hg a *right coset* of H in G .

Let's illustrate this using the *subgroup* H above. If the element g that we pick from G is also an element of H , then $Hg = H$ as illustrated below. Notice, too, that even though we get back the same *coset* H , the elements of H can be in different orders as a result of our multiplication. In general, that is to be expected.

$$H \cdot () = \left\{ \begin{array}{c} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} \cdot () = \left\{ \begin{array}{c} () \cdot () \\ (1,2,3) \cdot () \\ (1,3,2) \cdot () \end{array} \right\} = \left\{ \begin{array}{c} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} = H$$

$$H \cdot (1,2,3) = \left\{ \begin{array}{c} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} \cdot (1,2,3) = \left\{ \begin{array}{c} () \cdot (1,2,3) \\ (1,2,3) \cdot (1,2,3) \\ (1,3,2) \cdot (1,2,3) \end{array} \right\} = \left\{ \begin{array}{c} (1,2,3) \\ (1,3,2) \\ () \end{array} \right\} = H$$

$$H \cdot (1,3,2) = \left\{ \begin{array}{c} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} \cdot (1,3,2) = \left\{ \begin{array}{c} () \cdot (1,3,2) \\ (1,2,3) \cdot (1,3,2) \\ (1,3,2) \cdot (1,3,2) \end{array} \right\} = \left\{ \begin{array}{c} (1,3,2) \\ () \\ (1,2,3) \end{array} \right\} = H$$

On the other hand, if g is not an element of G , then we'll get something completely different from H .

$$H \cdot (1,2) = \left\{ \begin{array}{l} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} \cdot (1,2) = \left\{ \begin{array}{l} () \cdot (1,2) \\ (1,2,3) \cdot (1,2) \\ (1,3,2) \cdot (1,2) \end{array} \right\} = \left\{ \begin{array}{l} (1,2) \\ (2,3) \\ (1,3) \end{array} \right\}$$

$$H \cdot (1,3) = \left\{ \begin{array}{l} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} \cdot (1,3) = \left\{ \begin{array}{l} () \cdot (1,3) \\ (1,2,3) \cdot (1,3) \\ (1,3,2) \cdot (1,3) \end{array} \right\} = \left\{ \begin{array}{l} (1,3) \\ (1,2) \\ (2,3) \end{array} \right\}$$

$$H \cdot (2,3) = \left\{ \begin{array}{l} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} \cdot (2,3) = \left\{ \begin{array}{l} () \cdot (2,3) \\ (1,2,3) \cdot (2,3) \\ (1,3,2) \cdot (2,3) \end{array} \right\} = \left\{ \begin{array}{l} (2,3) \\ (1,3) \\ (1,2) \end{array} \right\}$$

And now, there's a whole bunch of important things to notice. First, we've found only two distinct *right cosets* that are given below.

$$H = H \cdot () = \left\{ \begin{array}{l} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\}$$

$$H \cdot (2,3) = \left\{ \begin{array}{l} (2,3) \\ (1,3) \\ (1,2) \end{array} \right\}$$

Of these *right cosets*, only H is a *subgroup* of G . We know immediately that $H \cdot (2,3)$ is not a *subgroup* of G since the set does not contain the identity element. However, do notice that each *right coset* contains the same number of elements and that every element of G belongs in one or the other of these two *right cosets*. In particular, that means that the total number of elements in G is equal to the number of elements in H times the number of *right cosets* of H in G . In this case, the equation comes out to $6=3 \times 2$. Furthermore, this sort of thing is going to happen all the time, and in *group theory* this result is known as *Lagrange's*

Theorem. Thus, in general, this means that for any *group* G with *subgroup* H , we have that “the number of elements in G ” equals “the number of elements in H ” times “the number of right cosets of H in G ”. And a spectacular and very important consequence of this is that the number of elements in a subgroup of a group G must be a divisor of the number of elements of G . Thus, in this example, the order or number of elements in G is six, the order or number of elements in H is three, and three divides evenly into six. In modern notation, we usually write it this way where $|G|$ represents the number of elements in the a group G :

$$|G|=6, |H|=3, \frac{|G|}{|H|}=2$$

Again, *Lagrange’s Theorem* is a very important result that says the order of a *subgroup* always divides the order of the *group*. However, the converse of *Lagrange’s Theorem* is not necessarily true. That is, just because, for example, 10 divides 20, that doesn’t mean that every *group* of order 20 has a *subgroup* of order 10. It only means that whatever *subgroups* of a *group* of order 20 exist, the only possible orders for them are 1, 2, 4, 5, 10, an 20 since these are the only natural numbers that divide evenly into 20 with no remainder. Nonetheless, there is another advanced theorem called the *Sylow Theorem* tells us that if a *prime number* raised to a power divides the order of a *group*, then our *group* does indeed contain a *subgroup* composed of that many elements. Thus, for example, a group G of order 8 must contain *subgroups* of orders $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, and $2^3 = 8$.

The second less spectacular but equally important thing to notice from our example is that if g_1 and g_2 belong to the same *right coset* of H in G , then $Hg_1 = Hg_2$. This is something that’ll always happen, and in a later part of this series we will give a proof of this fact. Additionally, this implies that distinct *cosets* never have any elements in common!

Now let's look at a second *subgroup* $H_1 = \langle (1,2) \rangle = \{(), (1,2)\}$ and its *right cosets* in $G = \{(), (2,3), (1,2), (1,2,3), (1,3,2), (1,3)\}$. When we do, we get the following distinct *cosets*:

$$H_1 = \left\{ \begin{array}{l} () \\ (1,2) \end{array} \right\}$$

$$H_1(1,3) = \left\{ \begin{array}{l} () \\ (1,2) \end{array} \right\} \cdot (1,3) = \left\{ \begin{array}{l} () \cdot (1,3) \\ (1,2) \cdot (1,3) \end{array} \right\} = \left\{ \begin{array}{l} (1,3) \\ (1,2,3) \end{array} \right\}$$

$$H_1(2,3) = \left\{ \begin{array}{l} () \\ (1,2) \end{array} \right\} \cdot (2,3) = \left\{ \begin{array}{l} () \cdot (2,3) \\ (1,2) \cdot (2,3) \end{array} \right\} = \left\{ \begin{array}{l} (2,3) \\ (1,3,2) \end{array} \right\}$$

Notice that once again we have that the set G is equal to the union of all three *right cosets*, written as $G = H_1 \cup [H_1 \cdot (1,3)] \cup [H_1 \cdot (2,3)]$, and that the order or number of elements in G , written as $|G|$, is given by:

$$|G| = 6 = 2 \cdot 3 = |H_1| \cdot (\text{the number of right cosets of } H \text{ in } G).$$

Again, the order of our *subgroup* is a divisor of the order of G , and since $|G| = 6$, this means that it's conceivable that G might have subgroups of orders, 1, 2, 3, and 6, but there is no way that it could have a subgroup of order 4 or 5. The order of a *subgroup* must always be a divisor of the order of the *group*!

Now let's do one more example to show how easy it is to construct all the *right cosets* if we are dealing with a *group* G that is not too terribly large. Thus, this time let's start with the following *subgroup* H_2 of our original *group* G with 6 elements:

$$H_2 = \left\{ \begin{pmatrix} () \\ (1,3) \end{pmatrix} \right\}$$

To construct a *right coset* of H_2 , just pick an element of G that hasn't been used yet such as $(1,2)$. This choice gives us the following result:

$$H_2(1,2) = \left\{ \begin{pmatrix} () \\ (1,3) \end{pmatrix} \right\} (1,2) = \left\{ \begin{pmatrix} () \cdot (1,2) \\ (1,3) \cdot (1,2) \end{pmatrix} \right\} = \left\{ \begin{pmatrix} (1,2) \\ (1,3,2) \end{pmatrix} \right\}$$

Our original *subgroup* H_2 and the subsequent *right coset* account now for 4, but not all, of the elements of G . Thus, we can continue. Just pick an element of G that hasn't been used yet, and construct yet another *right coset*. In this case, I pick $(2,3)$ and get the following:

$$H_2(2,3) = \left\{ \begin{pmatrix} () \\ (1,3) \end{pmatrix} \right\} (2,3) = \left\{ \begin{pmatrix} () \cdot (2,3) \\ (1,3) \cdot (2,3) \end{pmatrix} \right\} = \left\{ \begin{pmatrix} (2,3) \\ (1,2,3) \end{pmatrix} \right\}$$

And now, each element of G belongs to one of the three *cosets*, H_2 , $H_2(1,2)$, or $H_2(2,3)$, and we're done!

LEFT COSETS

Let's begin with the same multiplication table that we examined previously for the *group* G that is generated by the cycles $(1,2)$ and $(1,3)$.

```
gap> a:=(1,2);
(1,2)

gap> b:=(1,3);
(1,3)

gap> g:=Group(a,b);
Group([ (1,2), (1,3) ])

gap> Size(g);
6

gap> Elements(g);
[ (), (2,3), (1,2), (1,2,3), (1,3,2), (1,3) ]

gap> ShowMultiplicationTable(g);
*
-----
() | () (2,3) (1,2) (1,2,3) (1,3,2) (1,3)
(2,3) | (2,3) () (1,2,3) (1,2) (1,3) (1,3,2)
(1,2) | (1,2) (1,3,2) () (1,3) (2,3) (1,2,3)
(1,2,3) | (1,2,3) (1,3) (2,3) (1,3,2) () (1,2)
(1,3,2) | (1,3,2) (1,2) (1,3) () (1,2,3) (2,3)
(1,3) | (1,3) (1,2,3) (1,3,2) (2,3) (1,2) ()
```

If H is a *subgroup* of the *group* G , then we previously defined a *right coset* of H as $Hg = \{h \cdot g \mid h \in H\}$, where $g \in G$. Notice that we could just as easily have defined a *left coset* of H as $gH = \{g \cdot h \mid h \in H\}$, where $g \in G$. Furthermore, most textbooks have a preference for talking in terms of *left cosets* over *right cosets*. However, this preference is just a personal bias on the part of other authors, and since the software program GAP favors *right cosets* over *left cosets*, so will we. Nonetheless, let's compare some *right* versus *left cosets* for a couple of different *subgroups* of G . First we have the following:

$$G = \left\{ \begin{array}{l} () \\ (2,3) \\ (1,2) \\ (1,2,3) \\ (1,3,2) \\ (1,3) \end{array} \right\}$$

$$H = \left\{ \begin{array}{l} () \\ (1,2) \end{array} \right\}$$

$$H(1,3) = \left\{ \begin{array}{l} () \\ (1,2) \end{array} \right\} \cdot (1,3) = \left\{ \begin{array}{l} (1,3) \\ (1,2,3) \end{array} \right\}$$

$$(1,3)H = (1,3) \cdot \left\{ \begin{array}{l} () \\ (1,2) \end{array} \right\} = \left\{ \begin{array}{l} (1,3) \\ (1,3,2) \end{array} \right\}$$

The most important thing to observe in this result is that the above *right* and *left cosets* are not the same! And this is, indeed, what often happens. However, sometimes things work out otherwise. For example, let's take a different *subgroup* H of G and again examine the *right* and *left cosets* of H by element $(1,3)$.

$$H = \left\{ \begin{array}{l} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\}$$

$$H(1,3) = \left\{ \begin{array}{l} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} \cdot (1,3) = \left\{ \begin{array}{l} (1,3) \\ (1,2) \\ (3,2) \end{array} \right\}$$

$$(1,3)H = (1,3) \cdot \left\{ \begin{array}{l} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} = \left\{ \begin{array}{l} (1,3) \\ (2,3) \\ (1,2) \end{array} \right\}$$

In this case, we do get back the same result for both *cosets*, after noticing both that $(3,2) = (2,3)$ and that the elements appear in different orders in the two *cosets*. Furthermore, if we look at any other *right* and *left cosets* of H in G , the same thing would happen. The *right* and *left cosets* in this case are always the same, and when that happens with a *subgroup* like H , it is special and so we give H a special name. We call H a *normal subgroup* of G , and we write $H \triangleleft G$. We will point out that every *group* has at least two *normal subgroups*, itself and the *identity*. But if these are a *groups* only *normal subgroups*, then we call it *simple*.

QUOTIENT GROUPS

Recall that at the end of our chapter on *left cosets* we said that a *subgroup* H is called a *normal subgroup* of a *group* G if for every $g \in G$, we have that the *right coset* Hg is identical to the *left coset* gH . In other words, if $Hg = gH$ for every $g \in G$. We'll now reveal why *normal subgroups* are so important. It's because it's only if a *subgroup* is *normal* that are we assured that the *cosets* (either the *left* or the *right*) will form a *group* where the multiplication (for *right cosets*) is defined by $Ha \cdot Hb = H \cdot ab$. Let's illustrate this with the same multiplication table that we examined previously for the *group* G generated by the cycles $(1,2)$ and $(1,3)$.

```
gap> a:=(1,2);
(1,2)
gap> b:=(1,3);
(1,3)
gap> g:=Group(a,b);
Group([ (1,2), (1,3) ])
gap> Size(g);
6
gap> Elements(g);
[ (), (2,3), (1,2), (1,2,3), (1,3,2), (1,3) ]
gap> ShowMultiplicationTable(g);
*
-----
| () (2,3) (1,2) (1,2,3) (1,3,2) (1,3)
-----
() | () (2,3) (1,2) (1,2,3) (1,3,2) (1,3)
(2,3) | (2,3) () (1,2,3) (1,2) (1,3) (1,3,2)
(1,2) | (1,2) (1,3,2) () (1,3) (2,3) (1,2,3)
(1,2,3) | (1,2,3) (1,3) (2,3) (1,3,2) () (1,2)
(1,3,2) | (1,3,2) (1,2) (1,3) () (1,2,3) (2,3)
(1,3) | (1,3) (1,2,3) (1,3,2) (2,3) (1,2) ()
```

First, though, let's make just one quick definition that we'll need. In particular, if M and N are both subsets of a group G , then we'll define the product MN as $MN = \{mn \mid m \in M \text{ and } n \in N\}$. In other words, MN consists of all products we can make where the first factor comes from M and the second factor comes from N . Now let's do some calculations with the following subset H which is a *normal subgroup* of G .

$$H = \left\{ \begin{array}{l} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\}$$

We'll now compute both $H(1,3) \cdot H(1,2)$ and $H \cdot (1,3)(1,2)$ in order to show that the results are equal.

$$\begin{aligned} H(1,3) \cdot H(1,2) &= \left\{ \begin{array}{l} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} \cdot (1,3) \cdot \left\{ \begin{array}{l} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} \cdot (1,2) = \left\{ \begin{array}{l} (1,3) \\ (1,2) \\ (3,2) \end{array} \right\} \cdot \left\{ \begin{array}{l} (1,2) \\ (2,3) \\ (1,3) \end{array} \right\} \\ &= \left\{ \begin{array}{l} (1,3)(1,2) \\ (1,2)(1,2) \\ (3,2)(1,2) \\ (1,3)(2,3) \\ (1,2)(2,3) \\ (3,2)(2,3) \\ (1,3)(1,3) \\ (1,2)(1,3) \\ (3,2)(1,3) \end{array} \right\} = \left\{ \begin{array}{l} (1,3,2) \\ () \\ (3,1,2) \\ (1,3,2) \\ (1,3,2) \\ () \\ () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} = \left\{ \begin{array}{l} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} \end{aligned}$$

And similarly,

$$H \cdot (1,3)(1,2) = \left\{ \begin{array}{l} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} \cdot (1,3,2) = \left\{ \begin{array}{l} ()(1,3,2) \\ (1,2,3)(1,3,2) \\ (1,3,2)(1,3,2) \end{array} \right\} = \left\{ \begin{array}{l} (1,3,2) \\ () \\ (1,2,3) \end{array} \right\} = \left\{ \begin{array}{l} () \\ (1,2,3) \\ (1,3,2) \end{array} \right\} = H(1,3) \cdot H(1,2)$$

Thus, in at least this one example we have verified that $Ha \cdot Hb = H \cdot ab$ where H is a *normal subgroup* of G . Now let's see what happens in an example where H is not a *normal subgroup*. In particular, we'll let H be the following *subgroup*.

$$H = \left\{ \begin{pmatrix} () \\ (1,2) \end{pmatrix} \right\}$$

Let's look at $H(1,3) \cdot H(2,3)$ and $H \cdot (1,3)(2,3)$. On the one hand,

$$\begin{aligned} H(1,3) \cdot H(2,3) &= \left\{ \begin{pmatrix} () \\ (1,2) \end{pmatrix} \right\} \cdot (1,3) \cdot \left\{ \begin{pmatrix} () \\ (1,2) \end{pmatrix} \right\} \cdot (2,3) = \left\{ \begin{pmatrix} (1,3) \\ (1,2,3) \end{pmatrix} \right\} \cdot \left\{ \begin{pmatrix} (1,3) \\ (1,3,2) \end{pmatrix} \right\} \\ &= \left\{ \begin{pmatrix} (1,3)(1,3) \\ (1,2,3)(1,3) \\ (1,3)(1,3,2) \\ (1,2,3)(1,3,2) \end{pmatrix} \right\} = \left\{ \begin{pmatrix} () \\ (1,2) \\ (1,2) \\ () \end{pmatrix} \right\} = \left\{ \begin{pmatrix} () \\ (1,2) \end{pmatrix} \right\} \end{aligned}$$

But on the other hand,

$$H \cdot (1,3)(2,3) = \left\{ \begin{pmatrix} () \\ (1,2) \end{pmatrix} \right\} \cdot (1,2,3) = \left\{ \begin{pmatrix} (1,2,3) \\ (1,3) \end{pmatrix} \right\} \neq H(1,3) \cdot H(2,3)$$

Thus, in this instance we see that $Ha \cdot Hb \neq H \cdot ab$, and so we can't get a group by defining multiplication in this manner. However, if H were a *normal subgroup* of G , then this method of multiplication would give us a *group of cosets*, and we'll formally prove this in another part of this work. In closing, though, I'd like to mention that when we do have a *normal subgroup* such as $N \triangleleft G$, then the resulting *group of right (or left) cosets* is called either a *factor group* or a *quotient group*. In this new *group of cosets*, generally denoted by G/N , the identity element is the *normal subgroup* N , and we like to say that in this *quotient* or *factor group* G/N that the *subgroup* N has been factored out in the same way that the number 5 is factored out in the quotient $\frac{2 \cdot 5}{3 \cdot 5} = \frac{2}{3}$. In particular, in our *quotient group* we have "factored out" the differences between the individual elements of N so that all we see is N itself, but no longer its constituent parts. We'll talk more about this in the next chapter on *Equivalence Relations*.

EQUIVALENCE RELATIONS

One of the most fundamental relationships between numbers or expressions in mathematics is that of equality such as when we write $2=2$ or $3+1=4$. The equal sign, itself, was introduced in 1557 by the Welsh mathematician Robert Recorde who correctly noted that probably nothing could symbolize equality better than two identical parallel lines of equal length. In time, though, mathematicians noted that equality is characterized by three fundamental properties that we call *reflexive*, *symmetric*, and *transitive*.

$$a = a \quad (\text{reflexive property})$$

$$\text{If } a = b, \text{ then } b = a \quad (\text{symmetric property})$$

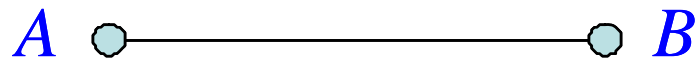
$$\text{If } a = b \text{ and } b = c, \text{ then } a = c \quad (\text{transitive property})$$

Abstracting from this (as mathematicians have a great tendency to do), it was decided that any sort of relationship between objects that bears these three properties would be similar to equality and should be called an *equivalence relation*. For example, let's let our objects be chairs, and let's say that two chairs are related if they were made in the same factory. Then this relationship is *reflexive* because it's true to say that chair A was made in the same factory as chair A . Likewise the relationship is *symmetric* because if chair A was made in the same factory as chair B , then it is obviously true that chair B was made in the same factory as chair A . And finally, the relationship is *transitive* because if chair A was made in the same factory as chair B and if chair B was made in the same factory as chair C , then clearly chair A was made in the same factory as chair C . A common notation these days that is used to say that " a is equivalent to b " is to write $a \sim b$. However, I prefer an older notation, $a \equiv b$, and I'll use this throughout.

Another way to look at *equivalence relations*, aside from the above three mentioned properties, is that an *equivalence relation* is like taking a set of objects

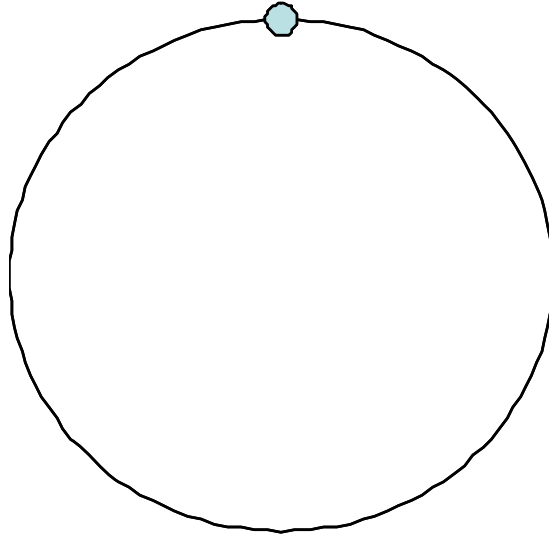
and then separating the objects into this or that box, and two of the objects are considered *equivalent* if they wind up in the same box. For example, let's let our objects be the set of *integers* $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$, and let's say that $a \equiv b$ if each integer results in the same remainder upon division by 2. Essentially, this results in the integers being divided and thrown into two separate boxes, one that we'll call the *even integers* and another that we'll call the *odd integers*. Furthermore, anytime you partition a set of objects by creating a set of boxes such that all the objects wind up in one of the boxes and no two boxes have any objects in common, that partition defines an *equivalence relation* and we call the contents of any particular box an *equivalence class*.

In higher mathematics, the notion of an *equivalence relation* is of the utmost importance because it is a primary way in which we construct new objects from old. For example, consider the line segment below.



If we see the two endpoints of this line segment as being *equivalent*, rather than separate, to one another, then our line segment is transformed into a closed loop. The bottom line is that we can actually change our perception of reality by virtue of what things we consider *equivalent* and what things we consider different. Thus, for instance, one person's notions of good and bad can create a perceived reality that is different from another person's. And in politics, these differences are what result in different political parties!

$$A = B$$



In *group theory*, anytime we have a *subgroup* of a *group* (*normal* or not), then the *right cosets* of that *subgroup* create a partition of the *group* that defines an *equivalence relation*, and each individual *coset* is an *equivalence class* in that relationship. We experienced this earlier with the following *group* G of six permutations and its *subgroup* H consisting of the two permutations $()$ and $(1,2)$. This setup partitioned G into three distinct *right cosets*, and the membership in a *coset* defines an *equivalence relation* within the group G .

$$G = \left\{ \begin{array}{l} () \\ (2,3) \\ (1,2) \\ (1,2,3) \\ (1,3,2) \\ (1,3) \end{array} \right\}$$

$$H = \left\{ \begin{array}{l} () \\ (1,2) \end{array} \right\} \quad H(1,3) = \left\{ \begin{array}{l} (1,3) \\ (1,2,3) \end{array} \right\} \quad H(2,3) = \left\{ \begin{array}{l} (2,3) \\ (1,3,2) \end{array} \right\}$$

As a reminder, though, recall that if our *subgroup* is *normal*, then the *right cosets* also define their own *group*. Additionally, you should realize that an *equivalence relation* is just another way of looking at a *quotient structure*. That is, for everything that falls into the same box or *equivalence class* the differences between those objects have been divided out just like when we “divide out” or eliminate the differences between specific *integers* when we see them as only *odd* or *even*. This is the essence of an *equivalence relation*, and this is also the essence of a *quotient structure* in higher mathematics.

SOMETHING FROM NOTHING CREATION

As mentioned previously, *equivalence relations* and *quotient structures* are just two ways of looking at the same thing. In each instance, things are being pigeonholed into their own boxes, and all those items that lie within the same box are considered the same as we no longer take notice of the individual differences. Again, this is what happens when we take the *integers*, $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$, and separate them into two groups, *odd* or *even*. At the *integer* level we have an infinite number of objects, but in the resulting *quotient structure* there are only two, *odd* or *even*, and the *odd integers* are no longer distinguished from one another at this level just as the individual *even integers* are also no longer distinguished.

This type of organization or *quotient structure* occurs every time we have an epiphany or “aha” moment. At such moments, we often speak in terms of the pieces of the puzzle coming together or of our brains suddenly connecting the dots. These are just metaphors, though, for the sudden realization that certain objects we’ve been perceiving belong together in some unique way, and by grouping them together we also create something new. We create a *quotient structure* where various objects are no longer seen as separate from one another, but rather as component pieces of a unified whole. And when we do this, a whole new world is created for us that contains a new perception or realization. And this is what I think is rightfully called *something from nothing creation* because our epiphanies generally seem to pop out of nothingness into our awareness. For example, at some point as a toddler we probably perceived a chair only as a collection of seemingly unrelated shapes and colors, not unlike an unassembled purchase from IKEA! But then, eventually, the differences between components were erased and we saw “chair” as a unified whole. In a similar manner, items must come together in unity for all of our realizations, and for any

artificial intelligence to convince me that it is sentient, it must also be able to construct *quotient structures* and have “aha” moments just as we do.

SPECIAL SUBGROUPS

We know at this point that if a *group* G is *abelian*, then $ab = ba$ for all $a, b \in G$. In other words, every element in G commutes with every other element in G . This latter statement, however, raises the question, “What if only some of the elements of G commute with every other element of G ?” First of all, we know that every group has at least one element that commutes with every other element, namely the *identity element*. And second, it turns out that for any *group* G the set of elements in G that commute with every other element of G form a *subgroup* of G called the *center of G* , and the standard notation for the *center* is $Z(G)$. The letter Z in this case stands for *zentrum*, the German word for *center*.

In the case of an *abelian group*, the *center* is automatically the entire *group* since every element commutes with every other element. However, if the *group* is not *abelian*, then the *center* could be the *identity element* or something in between the *identity* and the entire *group*. Below are a few examples created in GAP. Also, the *center of G* is additionally a *normal subgroup* of G .

```
gap> a:=(1, 2, 3, 4, 5);
(1, 2, 3, 4, 5)

gap> g:=Group(a);
Group([ (1, 2, 3, 4, 5) ])

gap> c:=Center(g);
Group([ (1, 2, 3, 4, 5) ])

gap> Elements(c);
[ (), (1, 2, 3, 4, 5), (1, 3, 5, 2, 4), (1, 4, 2, 5, 3), (1, 5, 4, 3, 2) ]

gap> a:=(1, 2, 3);
(1, 2, 3)

gap> b:=(2, 3);
(2, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3), (2, 3) ])

gap> c:=Center(g);
Group(())
```

```

gap> Elements(c);
[ () ]

gap> a:=(1, 2, 3, 4);
(1, 2, 3, 4)

gap> b:=(1, 3);
(1, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3, 4), (1, 3) ])

gap> c:=Center(g);
Group([ (1, 3)(2, 4) ])

gap> Elements(c);
[ (), (1, 3)(2, 4) ]

```

Now let's go back to the equation $ab=ba$ which is always true for elements of *abelian groups*. By moving elements to one side via multiplication by *inverses*, we can rewrite this equation as either $aba^{-1}b^{-1}=e$ or as $a^{-1}b^{-1}ab=e$. In *abelian groups*, the expressions $aba^{-1}b^{-1}$ and $a^{-1}b^{-1}ab$ always come out equal to the *identity*, but in *nonabelian groups* this will not always be the case. Consequently, we call expressions like $aba^{-1}b^{-1}$ and $a^{-1}b^{-1}ab$ *commutators*, and if our *group* is *nonabelian*, then the set of *commutators* for that *group* will generate a *subgroup* called the *commutator* or *derived subgroup*. Furthermore, the *commutator* or *derived subgroup* of G is a *normal subgroup* of G , and the corresponding *quotient* or *factor group* is always *abelian*. Again, for an *abelian group* the *commutator* or *derived subgroup* will consist only of the *identity*, but for a *nonabelian group* it will be less trivial. Here are a few derivations of the *commutator subgroup* created using GAP.

```

gap> a:=(1, 2, 3, 4, 5);
(1, 2, 3, 4, 5)

gap> g:=Group(a);
Group([ (1, 2, 3, 4, 5) ])

gap> d:=DerivedSubgroup(g);
Group(())

gap> Elements(d);
[ () ]

```

```

gap> a:=(1, 2, 3);
(1, 2, 3)

gap> b:=(2, 3);
(2, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3), (2, 3) ])

gap> d:=DerivedSubgroup(g);
Group([ (1, 3, 2) ])

gap> Elements(d);
[ (), (1, 2, 3), (1, 3, 2) ]

gap> a:=(1, 2, 3, 4);
(1, 2, 3, 4)

gap> b:=(1, 3);
(1, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3, 4), (1, 3) ])

gap> d:=DerivedSubgroup(g);
Group([ (1, 3)(2, 4) ])

gap> Elements(d);
[ (), (1, 3)(2, 4) ]

```

Now let's suppose we have a *group* G with a *subgroup* H such that H is a *normal subgroup* of G . That means that for every $a \in G$ we have that the *right coset* by a is equal to the *left coset* by a , $Ha = aH$. Another way to rewrite this is as either $a^{-1}Ha = H$ or $H = aHa^{-1}$. However, suppose that G has a *subgroup* H_1 that is not a *normal subgroup* of G . If that were the case, then there exists at least one $a \in G$ such that $H_1a \neq aH_1$, and it immediately follows that $a^{-1}H_1a \neq H_1$. However, what will be true is that there will be another *subgroup* H_2 such that $a^{-1}H_1a = H_2$. When this happens we say that the *subgroup* H_1 is a *conjugate* of the *subgroup* H_2 . Also, if H_1 is *conjugate* to H_2 , then H_1 will also be *isomorphic* to H_2 . Recall that that means that these two *subgroups* will not only have the same number of elements, but also that these elements may be paired from one *group* to the other in such a way that multiplication in one *group* corresponds to multiplication in the other *group*. In other words, their respective multiplication tables have the same structure. Below we show you how to find *conjugate subgroups* using GAP.

```

gap> a:=(1, 2, 3);
(1, 2, 3)

gap> b:=(2, 3);
(2, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3), (2, 3) ])

gap> h:=Group(b);
Group([ (2, 3) ])

gap> IsNormal(g, h);
false

gap> cs:=ConjugateSubgroups(g, h);
[ Group([ (2, 3) ]), Group([ (1, 2) ]), Group([ (1, 3) ]) ]

gap> Elements(cs[1]);
[ (), (2, 3) ]

gap> Elements(cs[2]);
[ (), (1, 2) ]

gap> Elements(cs[3]);
[ (), (1, 3) ]

```

In our chapter on *Right Cosets* we mentioned the *Sylow Theorem*, discovered by Norwegian mathematician Ludwig Sylow (1832-1918), that guarantees us that if p^n is the largest power of a prime p that divides the order of a group G , then G has a *subgroup* of order or size p^n . A *subgroup* of this sort is called a *Sylow p -subgroup*. Furthermore, if our *Sylow p -subgroup*, for a given prime, is not a *normal subgroup*, then there will be several *Sylow p -subgroups* for that prime and they will all be *conjugate* to one another. Additionally, given that p^n is the highest power of our prime p that divides the order of G , we can, thus, write the order of G as $|G|=p^n \cdot m$ and the number of *conjugates* of our *Sylow p -subgroup* will also be a divisor of m . Now here are some examples showing how to use GAP to find *Sylow p -subgroups*.

```

gap> s3:=SymmetricGroup(3);
Sym([ 1 .. 3 ])

gap> Size(s3);
6

gap> FactorsInt(6);
[ 2, 3 ]

gap> sylow3:=SylowSubgroup(s3, 3);
Group([ (1, 2, 3) ])

```

```
gap> IsNormal (s3, syl ow3);
true

gap> syl ow2: =Syl owSubgroup(s3, 2);
Group([ (1, 2) ])

gap> IsNormal (s3, syl ow2);
false

gap> c: =Conj ugateSubgroups(s3, syl ow2);
[ Group([ (1, 2) ]), Group([ (2, 3) ]), Group([ (1, 3) ]) ]

gap> Si ze(c);
3

gap> El ements(c[1]);
[ (), (1, 2) ]

gap> El ements(c[2]);
[ (), (2, 3) ]

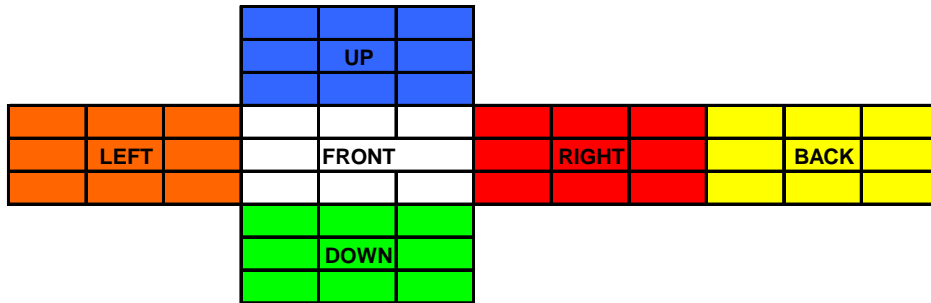
gap> El ements(c[3]);
[ (), (1, 3) ]
```

EXPLORING RUBIK'S CUBE WITH GAP

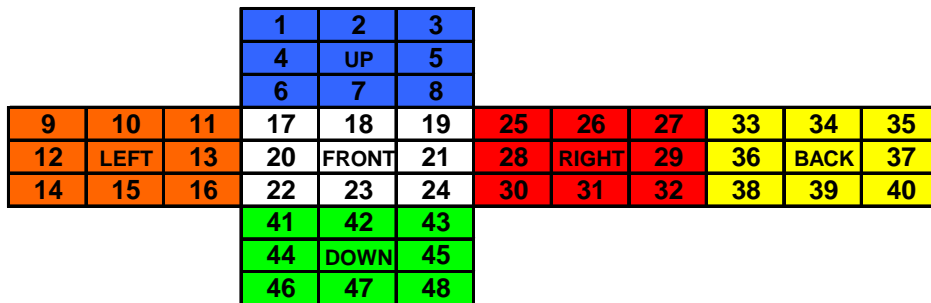
The software program known as GAP (*Groups, Algorithms, and Programming*) is a wonderful tool for exploring various aspects of Rubik's cube! Recall that the *Rubik's cube group* of permutations can be generated by the quarter rotations in the clockwise direction of the faces that we have called right, left, up, down, front, and back (R,L,U,P,F,B). In the GAP program, we will represent these moves by small, uncapitalized letters in order to make the typing easier. Also, if we want to keep track of the result of each rotation, then we can label each facelet of the cube with a number and then see how the numbers are permuted by each move. However, notice that we don't need to number the center facelets of each face because they always stay in the same position. Also notice that if we give the right face a quarter turn in the clockwise direction, then this will not only move facelets that are on the red face of the cube below, it will also move the facelets on the right side of the front, back, up, and down faces.



To help us understand the permutations created by the rotations of the various faces, we will unfold the cube and present it in 2-dimensions as we've done below.



And now we can easily give each *facelet* a number.



With this numbering, we can represent each possible clockwise quarter turn of a face of the cube by a unique permutation written in cycle notation, and to make it easy for you, here is the result below written just as you might enter it into GAP.

```
r:=(25,27,32,30)(26,29,31,28)(3,38,43,19)(5,36,45,21)(8,33,48,24);
l:=(9,11,16,14)(10,13,15,12)(1,17,41,40)(4,20,44,37)(6,22,46,35);
u:=(1,3,8,6)(2,5,7,4)(9,33,25,17)(10,34,26,18)(11,35,27,19);
d:=(41,43,48,46)(42,45,47,44)(14,22,30,38)(15,23,31,39)(16,24,32,40);
f:=(17,19,24,22)(18,21,23,20)(6,25,43,16)(7,28,42,13)(8,30,41,11);
b:=(33,35,40,38)(34,37,39,36)(3,9,46,32)(2,12,47,29)(1,14,48,27);
```

At this point, you might realize that if you want to use GAP to explore the consequences of these permutations, it would be a real pain to have to type and retype all of the above each time you want to use the program to explore Rubik's cube. And to complicate matters, since GAP is not really a Windows program,

the usual “save” and “paste” commands don’t work. Thus, this is what you can do to make things easy.

- First, copy the six lines of text above for r , l , u , d , f , and b exactly as given.
- Second, open up Notepad, and paste what you copied into Notepad.
- Third, save this file to your C-drive as *rubik.txt*.
- Fourth, open GAP.
- And lastly, enter the following code:

```
gap> Read("C: /rubi k. txt");
gap>
```

You won’t see anything spectacular on your screen after executing this last command, but, nonetheless, the definitions you’ve made for r , l , u , d , f , and b are now available for you to use. For example, here are a few calculations you can now do in GAP. The first computation displays the permutation r causes by rotating the right-hand face clockwise a quarter turn, and the subsequent lines show that r generates a cyclic *group* of order 4 that is *abelian*.

```
gap> r;
(3, 38, 43, 19) (5, 36, 45, 21) (8, 33, 48, 24) (25, 27, 32, 30) (26, 29, 31, 28)

gap> g1:=Group(r);
Group([ (3, 38, 43, 19) (5, 36, 45, 21) (8, 33, 48, 24) (25, 27, 32, 30) (26, 29, 31, 28) ])

gap> Si ze(g1);
4

gap> El ements(g1);
[ (), (3, 19, 43, 38) (5, 21, 45, 36) (8, 24, 48, 33) (25, 30, 32, 27) (26, 28, 31, 29),
(3, 38, 43, 19) (5, 36, 45, 21) (8, 33, 48, 24) (25, 27, 32,
30) (26, 29, 31, 28),
(3, 43) (5, 45) (8, 48) (19, 38) (21, 36) (24, 33) (25, 32) (26, 31) (27, 30) (28, 29) ]

gap> I sAbel i an(g1);
true
```

Next, we use GAP to examine the *group* generated by the product r times u . The result is a *cyclic group* containing 105 elements.

```

gap> g2:=Group(r*u);
Group([ (1, 3, 38, 43, 11, 35, 27, 32, 30, 17, 9, 33, 48, 24, 6) (2, 5, 36, 45, 21, 7, 4) (8, 25, 19) (1
0, 34, 26, 29, 31, 28, 18) ])
gap> Size(g2);
105
gap> IsAbelian(g2);
true

```

In contrast to this *abelian group*, the *group* generated by forming finite products of the elements r and u is *nonabelian* and contains 73,483,200 elements.

```

gap> g3:=Group(r,u);
<permutation group with 2 generators>
gap> Size(g3);
73483200
gap> IsAbelian(g3);
false

```

If we look instead at the *group* generated by r and l , turns of the right face and the left face of the cube, then the result this time is an *abelian group* of order 16.

```

gap> g4:=Group(r,l);
<permutation group with 2 generators>
gap> Size(g4);
16
gap> IsAbelian(g4);
true

```

Another *abelian group* can be generated by the elements r^2l^2 , u^2d^2 , and f^2b^2 , and this one has only 8 elements!

```

gap> g5:=Group(r^2*l^2, u^2*d^2, f^2*b^2);
<permutation group with 3 generators>
gap> Size(g5);
8
gap> IsAbelian(g5);
true

```


HOW TO USE GAP (PART 2)

For both convenience and continuity, we will always include the GAP commands presented in earlier parts of this book in black followed by the new commands which are printed in red.

1. *How can I redisplay the previous command in order to edit it?*

Press down on the control key and then also press p. In other words, “Ctrl p”.

2. *If the program gets in a loop and shows you the prompt “brk>” instead of “gap>”, how can I exit the loop?*

Press down on the control key and then also press d. In other words, “Ctrl d”.

3. *How can I exit the program?*

Either click on the “close” box for the window, or type “quit;” and press “Enter.”

4. *How do I find the inverse of a permutation?*

```
gap> a:=(1,2,3,4);  
(1,2,3,4)  
gap> a^-1;  
(1,4,3,2)
```

5. *How can I multiply permutations and raise permutations to powers?*

```
gap> (1,2)*(1,2,3);  
(1,3)
```

```
gap> (1,2,3)^2;  
(1,3,2)
```

```
gap> (1,2,3)^-1;  
(1,3,2)
```

```
gap> (1,2,3)^-2;  
(1,2,3)
```

```
gap> a:=(1,2,3);  
(1,2,3)
```

```
gap> b:=(1,2);  
(1,2)
```

```
gap> a*b;  
(2,3)
```

```
gap> a^2;  
(1,3,2)
```

```
gap> a^-2;  
(1,2,3)
```

```
gap> a^3;  
()
```

```
gap> a^-3;  
()
```

```
gap> (a*b)^2;  
()
```

```
gap> (a*b)^3;  
(2,3)
```

6. *How can I create a group from permutations, find the size of the group, and find the elements in the group?*

```
gap> a:=(1,2);  
(1,2)
```

```
gap> b:=(1,2,3);  
(1,2,3)
```

```
gap> g1:=Group(a,b);  
Group([ (1,2), (1,2,3) ])
```

```
gap> Size(g1);  
6
```

```
gap> Elements(g1);  
[ (), (2,3), (1,2), (1,2,3), (1,3,2), (1,3) ]
```

```
gap> g2:=Group([(1,2),(1,2,3)]);  
Group([ (1,2), (1,2,3) ])
```

```
gap> g3:=Group((1,2),(2,3,4));
Group([ (1,2), (2,3,4) ])
```

7. *How can I create a cyclic group of order 3?*

```
gap> a:=(1,2,3);
(1,2,3)
```

```
gap> g1:=Group(a);
Group([ (1,2,3) ])
```

```
gap> Size(g1);
3
```

```
gap> Elements(g1);
[ (), (1,2,3), (1,3,2) ]
```

```
gap> g2:=Group((1,2,3));
Group([ (1,2,3) ])
```

```
gap> g3:=CyclicGroup(IsPermGroup, 3);
Group([ (1, 2, 3) ])
```

8. *How can I create a multiplication table for the cyclic group of order 3 that I just created?*

```
gap> ShowMultiplicationTable(g1);
```

```
*      | ()      (1,2,3) (1,3,2)
-----+-----
()      | ()      (1,2,3) (1,3,2)
(1,2,3) | (1,2,3) (1,3,2) ()
(1,3,2) | (1,3,2) ()      1,2,3
```

9. *How do I determine if a group is abelian?*

```
gap> g1:=Group((1,2,3));
Group([ (1,2,3) ])
```

```
gap> IsAbelian(g1);
true
```

```
gap> g2:=Group((1,2),(1,2,3));
Group([ (1,2), (1,2,3) ])
```

```
gap> IsAbelian(g2);
false
```

10. *What do I type in order to get help for a command like “Elements?”*

```
gap> ?Elements
```

11. *How do I find all subgroups of a group?*

```
gap> a:=(1, 2, 3);
(1, 2, 3)
```

```
gap> b:=(2, 3);
(2, 3)
```

```
gap> g:=Group(a, b);
Group([ (1, 2, 3), (2, 3) ])
```



```

gap> Size(g);
6

gap> Elements(g);
[ () , (2, 3) , (1, 2) , (1, 2, 3) , (1, 3, 2) , (1, 3) ]

gap> h:=AllSubgroups(g);
[ Group(()) , Group([ (2, 3) ]) , Group([ (1, 2) ]) , Group([ (1, 3) ]) ,
Group([ (1, 2, 3) ]) , Group([ (1, 2, 3) , (2, 3) ]) ]

gap> List(h, i->Elements(i));
[ [ () ] , [ () , (2, 3) ] , [ () , (1, 2) ] , [ () , (1, 3) ] , [ () , (1, 2, 3) ,
(1, 3, 2) ] , [ () , (2, 3) , (1, 2) , (1, 2, 3) , (1, 3, 2) , (1, 3) ] ]

gap> Elements(h[1]);
[ () ]

gap> Elements(h[2]);
[ () , (2, 3) ]

gap> Elements(h[3]);
[ () , (1, 2) ]

gap> Elements(h[4]);
[ () , (1, 3) ]

gap> Elements(h[5]);
[ () , (1, 2, 3) , (1, 3, 2) ]

gap> Elements(h[6]);
[ () , (2, 3) , (1, 2) , (1, 2, 3) , (1, 3, 2) , (1, 3) ]

```

12. How do I find the subgroup generated by particular permutations?

```

gap> g:=Group((1, 2), (1, 2, 3));
Group([ (1, 2) , (1, 2, 3) ])

gap> Elements(g);
[ () , (2, 3) , (1, 2) , (1, 2, 3) , (1, 3, 2) , (1, 3) ]

gap> h:=Subgroup(g, [(1, 2)]);
Group([ (1, 2) ])

gap> Elements(h);
[ () , (1, 2) ]

```

13. How do I determine if a subgroup is normal?

```

gap> g:=Group((1, 2), (1, 2, 3));
Group([ (1, 2) , (1, 2, 3) ])

gap> h1:=Group((1, 2));
Group([ (1, 2) ])
gap> IsNormal(g, h1);

gap> h2:=Group((1, 2, 3));
Group([ (1, 2, 3) ])

```

```
gap> IsNormal (g, h2);
true
```

14. How do I find all normal subgroups of a group?

```
gap> g:=Group((1, 2), (1, 2, 3));
Group([ (1, 2), (1, 2, 3) ])

gap> Elements(g);
[ (), (2, 3), (1, 2), (1, 2, 3), (1, 3, 2), (1, 3) ]

gap> n:=NormalSubgroups(g);
[ Group([ (1, 2), (1, 2, 3) ]), Group([ (1, 3, 2) ]), Group(()) ]

gap> Elements(n[1]);
[ (), (2, 3), (1, 2), (1, 2, 3), (1, 3, 2), (1, 3) ]

gap> Elements(n[2]);
[ (), (1, 2, 3), (1, 3, 2) ]

gap> Elements(n[3]);
[ () ]
```

15. How do I determine if a group is simple?

```
gap> g:=Group((1, 2), (1, 2, 3));
Group([ (1, 2), (1, 2, 3) ])

gap> Elements(g);
[ (), (2, 3), (1, 2), (1, 2, 3), (1, 3, 2), (1, 3) ]

gap> IsSimple(g);
false

gap> h:=Group((1, 2));
Group([ (1, 2) ])

gap> Elements(h);
[ (), (1, 2) ]

gap> IsSimple(h);
true
```

16. How do I find the right cosets of a subset H of G?

```
gap> g:=Group([(1, 2, 3), (1, 2)]);
Group([ (1, 2, 3), (1, 2) ])
```

```

gap> Elements(g);
[ () , (2, 3) , (1, 2) , (1, 2, 3) , (1, 3, 2) , (1, 3) ]

gap> h:=Subgroup(g, [(1, 2)]);
Group([ (1, 2) ])

gap> Elements(h);
[ () , (1, 2) ]

gap> c:=RightCosets(g, h);
[ RightCoset(Group([ (1, 2) ]), ()) , RightCoset(Group([ (1, 2) ]), (1, 3, 2)) ,
RightCoset(Group([ (1, 2) ]), (1, 2, 3)) ]

gap> List(c, i->Elements(i));
[ [ () , (1, 2) ] , [ (2, 3) , (1, 3, 2) ] , [ (1, 2, 3) , (1, 3) ] ]
gap> Elements(c[1]);
[ () , (1, 2) ]

gap> Elements(c[2]);
[ (2, 3) , (1, 3, 2) ]

gap> Elements(c[3]);
[ (1, 2, 3) , (1, 3) ]

gap> rc:=RightCoset(h, (1, 2, 3));
RightCoset(Group([ (1, 2) ]), (1, 2, 3))

gap> Elements(rc);
[ (1, 2, 3) , (1, 3) ]

gap> rc:=h*(1, 2, 3);
RightCoset(Group([ (1, 2) ]), (1, 2, 3))

gap> Elements(rc);
[ (1, 2, 3) , (1, 3) ]

```

17. How can I create a quotient (factor) group?

```

gap> g:=Group([(1, 2, 3), (1, 2)]);
Group([ (1, 2, 3) , (1, 2) ])

gap> Elements(g);
[ () , (2, 3) , (1, 2) , (1, 2, 3) , (1, 3, 2) , (1, 3) ]

gap> n:=Group((1, 2, 3));
Group([ (1, 2, 3) ])

gap> Elements(n);
[ () , (1, 2, 3) , (1, 3, 2) ]

gap> IsNormal(g, n);
true

gap> c:=RightCosets(g, n);
[ RightCoset(Group([ (1, 2, 3) ]), ()) , RightCoset(Group([ (1, 2, 3) ]), (2, 3)) ]

gap> Elements(c[1]);
[ () , (1, 2, 3) , (1, 3, 2) ]

gap> Elements(c[2]);
[ (2, 3) , (1, 2) , (1, 3) ]

gap> f:=FactorGroup(g, n);
Group([ f1 ])

```

```

gap> Elements(f);
[ <identity> of ..., f1 ]

gap> ShowMultiplicationTable(f);
*          | <identity> of ... f1
-----+-----
<identity> of ... | <identity> of ... f1
f1              | f1          <identity> of ...

```

18. How do I find the center of a group?

```

gap> a:=(1, 2, 3);
(1, 2, 3)

gap> b:=(2, 3);
(2, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3), (2, 3) ])

gap> Center(g);
Group(())

gap> c:=Center(g);
Group(())

gap> Elements(c);
[ () ]

```

```

gap> a:=(1, 2, 3, 4);
(1, 2, 3, 4)

gap> b:=(1, 3);
(1, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3, 4), (1, 3) ])

gap> c:=Center(g);
Group([ (1, 3)(2, 4) ])

gap> Elements(c);
[ (), (1, 3)(2, 4) ]

```

19. How do I find the commutator (derived) subgroup of a group?

```

gap> a:=(1, 2, 3);
(1, 2, 3)

gap> b:=(2, 3);
(2, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3), (2, 3) ])

gap> d:=DerivedSubgroup(g);
Group([ (1, 3, 2) ])

```

```

gap> Elements(d);
[ (), (1, 2, 3), (1, 3, 2) ]

gap> a:=(1, 2, 3, 4);
(1, 2, 3, 4)

gap> b:=(1, 3);
(1, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3, 4), (1, 3) ])

gap> d:=DerivedSubgroup(g);
Group([ (1, 3)(2, 4) ])

gap> Elements(d);
[ (), (1, 3)(2, 4) ]

```

20. How do I find all Sylow p -subgroups for a given group?

```

gap> a:=(1, 2, 3);
(1, 2, 3)

gap> b:=(2, 3);
(2, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3), (2, 3) ])

gap> Size(g);
6

gap> FactorsInt(6);
[ 2, 3 ]

gap> sylow2:=SylowSubgroup(g, 2);
Group([ (2, 3) ])

gap> IsNormal(g, sylow2);
false

gap> c:=ConjugateSubgroups(g, sylow2);
[ Group([ (2, 3) ]), Group([ (1, 3) ]), Group([ (1, 2) ]) ]

gap> Elements(c[1]);
[ (), (2, 3) ]

gap> Elements(c[2]);
[ (), (1, 3) ]

gap> Elements(c[3]);
[ (), (1, 2) ]

gap> sylow3:=SylowSubgroup(g, 3);
Group([ (1, 2, 3) ])

gap> IsNormal(g, sylow3);
true

gap> Elements(sylow3);
[ (), (1, 2, 3), (1, 3, 2) ]

```

21. How can I create the Rubik's cube group using GAP?

First you need to save the following permutations as a pure text file with the name rubik.txt to your C-drive before you can import it into GAP.

```
r:=(25,27,32,30)(26,29,31,28)(3,38,43,19)(5,36,45,21)(8,33,48,24);
l:=(9,11,16,14)(10,13,15,12)(1,17,41,40)(4,20,44,37)(6,22,46,35);
u:=(1,3,8,6)(2,5,7,4)(9,33,25,17)(10,34,26,18)(11,35,27,19);
d:=(41,43,48,46)(42,45,47,44)(14,22,30,38)(15,23,31,39)(16,24,32,40);
f:=(17,19,24,22)(18,21,23,20)(6,25,43,16)(7,28,42,13)(8,30,41,11);
b:=(33,35,40,38)(34,37,39,36)(3,9,46,32)(2,12,47,29)(1,14,48,27);
```

And now you can read the file into GAP and begin exploring.

```
gap> Read("C:/rubik.txt");
gap> rubik:=Group(r,l,u,d,f,b);
<permutation group with 6 generators>
gap> Size(rubik);
43252003274489856000
```

22. How can I find the center of the Rubik's cube group?

```
gap> c:=Center(rubik);
Group([ (2,34)(4,10)(5,26)(7,18)(12,37)(13,20)(15,44)(21,28)(23,42)(29,36)(31,45)(39,47) ])
gap> Size(c);
2
gap> Elements(c);
[ (), (2,34)(4,10)(5,26)(7,18)(12,37)(13,20)(15,44)(21,28)(23,42)(29,36)(31,45)(39,47) ]
```

23. How can I find the commutator (derived) subgroup of the Rubik's cube group?

```
gap> d:=DerivedSubgroup(rubik);
<permutation group with 5 generators>
gap> Size(d);
21626001637244928000
gap> IsNormal(rubik,d);
true
```


false

```
gap> IsNormal (rubik, sylow11);  
false
```

NOTE: All of the *Sylow p -subgroups* found above have *conjugates*, but the sheer size of the *Rubik's cube group* makes it too difficult to pursue them on a typical desktop computer.

SUMMARY (PART 2)

In this second part we have already covered an amazing amount of information. In particular, you should now be able to explain the following concepts and notations.

- Subgroup
- Right coset
- Left coset
- Lagrange's Theorem
- Normal subgroup
- Simple subgroup
- $H \leq G$
- $H < G$
- $H \triangleleft G$
- $a \in G$
- Ha
- aH
- Quotient (Factor) groups
- Equivalence relation
 - Reflexive, symmetric, transitive
- Something from nothing creation
- \mathbb{Z}
- Center of a group, $Z(G)$
- Commutators
- Commutator or Derived subgroup
- Sylow p -subgroup
- Conjugate subgroups
- Creating and reading into GAP the rubik.txt file
- Using GAP to explore concepts introduced in part 2

PRACTICE (PART 2)

In many instances, do the problems below first by hand, and then check using GAP.

1. The permutations $(1,3)$ and $(1,2,3,4)$ generate a group G of eight elements. Use GAP to find the multiplication table for this group.
2. Using the multiplication table in problem 1, find all ten subgroups of G . (HINT: Begin with the cyclic subgroups.)
3. Classify each subgroup found in problem 2 as either cyclic or noncyclic.
4. Classify each subgroup found in problem 2 as either abelian or nonabelian.
5. Classify each subgroup found in problem 2 as either a normal or a not normal subgroup of G .
6. Find all simple subgroups of G .
7. The subgroup $N = \{ (), (1,3)(2,4), (1,2)(3,4), (1,4)(2,3) \}$ is normal in G . Find the right cosets of N in G , and construct a multiplication table for the quotient group (factor group) G/N .
8. For this problem, use GAP software and execute the command `Read("C:/rubik.txt");` before continuing. Next, find the size of the subgroup of the Rubik's cube group that is generated by $(r^2u^2)^3$. Also, do this operation on your Rubik's cube, and observe the pattern created.

9. For this problem, use GAP software and execute the command `Read("C:/rubik.txt");` before continuing. Next, find the size of the subgroup of the Rubik's cube group that is generated by $ud^{-1}rl^{-1}fb^{-1}ud^{-1}$. Also, do this operation on your Rubik's cube, and observe the pattern created.
10. For this problem, use GAP software and execute the command `Read("C:/rubik.txt");` before continuing. Next, find the size of the subgroup of the Rubik's cube group that is generated by r^2l^2 , u^2d^2 , and f^2b^2 . In other words, find the size of $H = \langle r^2l^2, u^2d^2, f^2b^2 \rangle$. Also, determine if H is abelian, and determine if H is a normal subgroup of the Rubik cube's group.
11. On paper, create two boxes – one for Oppose or Strongly Oppose and the other for Favor or Strongly Favor. Next, create an equivalence relation by placing each of the following phrases into only one of the boxes. When done, compare your answers with those of others or with mine and see how what you favor and what you oppose helps create your own personal reality?

Abortion is a woman's unrestricted right

Comfortable with same-sex marriage

Keep God in the public sphere

Privatize Social Security

Stricter punishment reduces crime

Absolute right to gun ownership

Higher taxes on the wealthy

Support American Exceptionalism

Expand the military

Prioritize green energy

Marijuana is a gateway drug

12. For this problem, use GAP software and execute the command `Read("C:/rubik.txt");` before continuing. Next, let $H = \langle r, u \rangle$, the subgroup of the Rubik's cube group generated by the permutations r and u . And now, find the size of H , the center of H , the size of the center of H , the elements in the center of H , and lastly find the commutator (derived) subgroup of H and the size of this commutator.

PRACTICE (PART 2) - ANSWERS

In many instances, do the problems below first by hand, and then check using GAP.

- The permutations $(1,3)$ and $(1,2,3,4)$ generate a group G of eight elements. Use GAP to find the multiplication table for this group.

```
gap> g:=Group((1,3),(1,2,3,4));
Group([ (1,3), (1,2,3,4) ])
gap> Size(g);
8
gap> ShowMultiplicationTable(g);
      |  ()  (2,4)  (1,2)(3,4)  (1,2,3,4)  (1,3)  (1,3)(2,4)  (1,4,3,2)  (1,4)(2,3)
-----|-----
()     |  ()  (2,4)  (1,2)(3,4)  (1,2,3,4)  (1,3)  (1,3)(2,4)  (1,4,3,2)  (1,4)(2,3)
(2,4)  |  (2,4)  ()  (1,2)(3,4)  (1,2,3,4)  (1,3)(2,4)  (1,3)  (1,4)(2,3)  (1,4,3,2)
(1,2)(3,4) |  (1,2)(3,4)  (1,4,3,2)  ()  (1,3)  (1,2,3,4)  (1,4)(2,3)  (2,4)  (1,3)(2,4)
(1,2,3,4) |  (1,2,3,4)  (1,4)(2,3)  (2,4)  (1,3)(2,4)  (1,2)(3,4)  (1,4,3,2)  ()  (1,3)
(1,3)    |  (1,3)  (1,3)(2,4)  (1,4,3,2)  (1,4)(2,3)  ()  (2,4)  (1,2)(3,4)  (1,2,3,4)
(1,3)(2,4) |  (1,3)(2,4)  (1,3)  (1,4)(2,3)  (1,4,3,2)  (2,4)  ()  (1,2,3,4)  (1,2)(3,4)
(1,4,3,2) |  (1,4,3,2)  (1,2)(3,4)  (1,3)  ()  (1,4)(2,3)  (1,2,3,4)  (1,3)(2,4)  (2,4)
(1,4)(2,3) |  (1,4)(2,3)  (1,2,3,4)  (1,3)(2,4)  (2,4)  (1,4,3,2)  (1,2)(3,4)  (1,3)  ()
```

- Using the multiplication table in problem 1, find all ten subgroups of G , and give the order of each subgroup. (HINT: Begin with the cyclic subgroups.)

$$\langle () \rangle = \{ () \}, \quad 1$$

$$\langle (2,4) \rangle = \{ (), (2,4) \}, \quad 2$$

$$\langle (1,2)(3,4) \rangle = \{ (), (1,2)(3,4) \}, \quad 2$$

$$\langle (1,2,3,4) \rangle = \{ (), (1,2,3,4), (1,3)(2,4), (4,3,2,1) \}, \quad 4$$

$$\langle (1,3) \rangle = \{ (), (1,3) \}, \quad 2$$

$$\langle (1,3)(2,4) \rangle = \{ (), (1,3)(2,4) \}, \quad 2$$

$$\langle (4,3,2,1) \rangle = \{ (), (4,3,2,1), (1,3)(2,4), (1,2,3,4) \}, \quad \text{Disregard, same group as } \langle (1,2,3,4) \rangle$$

$$\langle (1,4)(2,3) \rangle = \{ (), (1,4)(2,3) \}, \quad 2$$

$$\langle (1,3), (1,2,3,4) \rangle = \{ (), (2,4), (1,2)(3,4), (1,2,3,4), (1,3), (2,4), (4,3,2,1), (1,4)(2,3) \}, \quad 8$$

$$\langle (1,3), (2,4) \rangle = \{ (), (1,3), (2,4), (1,3)(2,4) \}, \quad 4$$

$$\langle (1,2)(3,4), (1,4)(2,3) \rangle = \{ (), (1,2)(3,4), (1,4)(2,3), (1,3)(2,4) \}, \quad 4$$

3. Classify each *subgroup* found in problem 2 as either *cyclic* or *noncyclic*.

$$\langle () \rangle = \{ () \} \quad \text{cyclic}$$

$$\langle (2,4) \rangle = \{ (), (2,4) \} \quad \text{cyclic}$$

$$\langle (1,2)(3,4) \rangle = \{ (), (1,2)(3,4) \} \quad \text{cyclic}$$

$$\langle (1,2,3,4) \rangle = \{ (), (1,2,3,4), (1,3)(2,4), (4,3,2,1) \} \quad \text{cyclic}$$

$$\langle (1,3) \rangle = \{ (), (1,3) \} \quad \text{cyclic}$$

$$\langle (1,3)(2,4) \rangle = \{ (), (1,3)(2,4) \} \quad \text{cyclic}$$

$$\langle (1,4)(2,3) \rangle = \{ (), (1,4)(2,3) \} \quad \text{cyclic}$$

$$\langle (1,3), (1,2,3,4) \rangle = \{ (), (2,4), (1,2)(3,4), (1,2,3,4), (1,3), (2,4), (4,3,2,1), (1,4)(2,3) \} \quad \text{noncyclic}$$

$$\langle (1,3), (2,4) \rangle = \{ (), (1,3), (2,4), (1,3)(2,4) \} \quad \text{noncyclic}$$

$$\langle (1,2)(3,4), (1,4)(2,3) \rangle = \{ (), (1,2)(3,4), (1,4)(2,3), (1,3)(2,4) \} \quad \text{noncyclic}$$

4. Classify each *subgroup* found in problem 2 as either *abelian* or *nonabelian*.

$$\langle () \rangle = \{ () \} \quad \text{abelian}$$

$$\langle (2,4) \rangle = \{ (), (2,4) \} \quad \text{abelian}$$

$$\langle (1,2)(3,4) \rangle = \{ (), (1,2)(3,4) \} \quad \text{abelian}$$

$$\langle (1,2,3,4) \rangle = \{ (), (1,2,3,4), (1,3)(2,4), (4,3,2,1) \} \quad \text{abelian}$$

$$\langle (1,3) \rangle = \{ (), (1,3) \} \quad \text{abelian}$$

$$\langle (1,3)(2,4) \rangle = \{ (), (1,3)(2,4) \} \quad \text{abelian}$$

$$\langle (1,4)(2,3) \rangle = \{ (), (1,4)(2,3) \} \quad \text{abelian}$$

$$\langle (1,3), (1,2,3,4) \rangle = \{ (), (2,4), (1,2)(3,4), (1,2,3,4), (1,3), (2,4), (4,3,2,1), (1,4)(2,3) \} \quad \text{nonabelian}$$

$$\langle (1,3), (2,4) \rangle = \{ (), (1,3), (2,4), (1,3)(2,4) \} \quad \text{abelian}$$

$$\langle (1,2)(3,4), (1,4)(2,3) \rangle = \{ (), (1,2)(3,4), (1,4)(2,3), (1,3)(2,4) \} \quad \text{abelian}$$

5. Classify each *subgroup* found in problem 2 as either a *normal* or a *not normal subgroup* of G .

$$\langle () \rangle = \{ () \} \quad \text{normal}$$

$$\langle (2,4) \rangle = \{ (), (2,4) \} \quad \text{not normal}$$

$$\langle (1,2)(3,4) \rangle = \{ (), (1,2)(3,4) \} \quad \text{not normal}$$

$$\langle (1,2,3,4) \rangle = \{ (), (1,2,3,4), (1,3)(2,4), (4,3,2,1) \} \quad \text{normal}$$

$$\langle (1,3) \rangle = \{ (), (1,3) \} \quad \text{not normal}$$

$$\langle (1,3)(2,4) \rangle = \{ (), (1,3)(2,4) \} \quad \text{normal}$$

$$\langle (1,4)(2,3) \rangle = \{ (), (1,4)(2,3) \} \quad \text{not normal}$$

$$\langle (1,3), (1,2,3,4) \rangle = \{ (), (2,4), (1,2)(3,4), (1,2,3,4), (1,3), (2,4), (4,3,2,1), (1,4)(2,3) \} \quad \text{normal}$$

$$\langle (1,3), (2,4) \rangle = \{ (), (1,3), (2,4), (1,3)(2,4) \} \quad \text{normal}$$

$$\langle (1,2)(3,4), (1,4)(2,3) \rangle = \{ (), (1,2)(3,4), (1,4)(2,3), (1,3)(2,4) \} \quad \text{normal}$$

6. Find all *simple subgroups* of G .

$$\langle () \rangle = \{ () \} \quad \text{simple}$$

$$\langle (2,4) \rangle = \{ (), (2,4) \} \quad \text{simple}$$

$$\langle (1,2)(3,4) \rangle = \{ (), (1,2)(3,4) \} \quad \text{simple}$$

$$\langle (1,2,3,4) \rangle = \{ (), (1,2,3,4), (1,3)(2,4), (4,3,2,1) \} \quad \text{not simple}$$

$$\langle (1,3) \rangle = \{ (), (1,3) \} \quad \text{simple}$$

$$\langle (1,3)(2,4) \rangle = \{ (), (1,3)(2,4) \} \quad \text{simple}$$

$$\langle (1,4)(2,3) \rangle = \{ (), (1,4)(2,3) \} \quad \text{simple}$$

$$\langle (1,3), (1,2,3,4) \rangle = \{ (), (2,4), (1,2)(3,4), (1,2,3,4), (1,3), (2,4), (4,3,2,1), (1,4)(2,3) \} \quad \text{not simple}$$

$$\langle (1,3), (2,4) \rangle = \{ (), (1,3), (2,4), (1,3)(2,4) \} \quad \text{not simple}$$

$$\langle (1,2)(3,4), (1,4)(2,3) \rangle = \{ (), (1,2)(3,4), (1,4)(2,3), (1,3)(2,4) \} \quad \text{not simple}$$

7. The subgroup $N = \{ (), (1,3)(2,4), (1,2)(3,4), (1,4)(2,3) \}$ is normal in G . Find the right cosets of N in G , and construct a multiplication table for the quotient group (factor group) G/N .

$$N = \left\{ \begin{array}{l} () \\ (1,3)(2,4) \\ (1,2)(3,4) \\ (1,4)(2,3) \end{array} \right\} \quad N \cdot (1,3) = \left\{ \begin{array}{l} () \\ (1,3)(2,4) \\ (1,2)(3,4) \\ (1,4)(2,3) \end{array} \right\} (1,3) = \left\{ \begin{array}{l} (1,3) \\ (2,4) \\ (1,2,3,4) \\ (4,3,2,1) \end{array} \right\}$$

	N	N(1,3)
N	N	N(1,3)
N(1,3)	N(1,3)	N

8. For this problem, use GAP software and execute the command `Read("C:/rubik.txt");` before continuing. Next, find the size of the subgroup of the Rubik's cube group that is generated by $(r^2u^2)^3$. Also, do this operation on your Rubik's cube, and observe the pattern created.


```
gap> h1:=Group((r^2*u^2)^3);
Group([ (2, 7)(18, 34)(21, 36)(28, 29) ])
gap> Size(h1);
2
```



9. For this problem, use GAP software and execute the command `Read("C:/rubik.txt");` before continuing. Next, find the size of the *subgroup* of the *Rubik's cube group* that is generated by $ud^{-1}rl^{-1}fb^{-1}ud^{-1}$. Also, do this operation on your Rubik's cube, and observe the pattern created.

```
gap> h2:=Group(u*d^-1*r*I^-1*f*b^-1*u*d^-1);
Group([ (1, 32, 22)(2, 31, 20)(3, 30, 17)(4, 29, 23)(5, 28, 18)(6, 27, 24)(7, 26, 21)(8, 25, 19)
(9, 38, 41)(10, 36, 42)(11, 33, 43)
(12, 39, 44)(13, 34, 45)(14, 40, 46)(15, 37, 47)(16, 35, 48) ])
gap> Size(h2);
3
```



10. For this problem, use GAP software and execute the command `Read("C:/rubik.txt");` before continuing. Next, find the size of the *subgroup* of the *Rubik's cube group* that is generated by r^2l^2 , u^2d^2 , and f^2b^2 . In other words, find the size of $H = \langle r^2l^2, u^2d^2, f^2b^2 \rangle$. Also, determine if H is *abelian*, and determine if H is a *normal subgroup* of the *Rubik cube's group*.

```
gap> H:=Group(r^2*l^2, u^2*d^2, f^2*b^2);
<permutation group with 3 generators>

gap> Size(H);
8

gap> IsAbelian(H);
true

gap> IsNormal(rubik, H);
false
```

11. On paper, create two boxes – one for *Oppose or Strongly Oppose* and the other for *Favor or Strongly Favor*. Next, create an *equivalence relation* by placing each of the following phrases into only one of the boxes. When done, compare your answers with those of others or with mine and see how what

you favor and what you oppose helps create your own personal reality?

Abortion is a woman's unrestricted right

Comfortable with same-sex marriage

Keep God in the public sphere

Privatize Social Security

Stricter punishment reduces crime

Absolute right to gun ownership

Higher taxes on the wealthy

Support American Exceptionalism

Expand the military

Prioritize green energy

Marijuana is a gateway drug

<u>OPPOSE OR STRONGLY OPPOSE</u>	<u>FAVOR OR STRONGLY FAVOR</u>
Keep God in the public sphere	Abortion is a woman's unrestricted right
Privatize Social Security	Comfortable with same-sex marriage
Stricter punishment reduces crime	Higher taxes on the wealthy
Absolute right to gun ownership	Prioritize green energy
Support American Exceptionalism	
Expand the military	
Marijuana is a gateway drug	

12. For this problem, use GAP software and execute the command `Read("C:/rubik.txt");` before continuing. Next, let $H = \langle r, u \rangle$, the subgroup of the Rubik's cube group generated by the permutations r and u . And now, find the size of H , the center of H , the size of the center of H , the elements in the center of H , and lastly find the commutator (derived) subgroup of H and the size of this commutator.

```
gap> Read("C:/rubik.txt");
gap> h:=Group(r,u);
<permutation group with 2 generators>
gap> Size(h);
73483200
gap> c:=Center(h);
Group([ (1, 9, 35) (3, 33, 27) (6, 17, 11) (8, 25, 19) (24, 30, 43) (32, 38, 48) ])
gap> Size(c);
3
gap> Elements(c);
[ (), (1, 9, 35) (3, 33, 27) (6, 17, 11) (8, 25, 19) (24, 30, 43) (32, 38, 48),
(1, 35, 9) (3, 27, 33) (6, 11, 17) (8, 19, 25) (24, 43, 30) (32, 48, 38) ]
gap> d:=DerivedSubgroup(h);
<permutation group with 5 generators>
gap> Size(d);
36741600
```



Real Women Know Group Theory!