

# Day 19

Tuesday June 19, 2012

## 1 Counting

We have talked about what it means for two things to have the same size; it is that we have a bijection between the one set and the other. Now we will delve into the idea of finite sets, and how to count the number of elements in a finite set.

This is an extremely useful concept. In the ever growing field of computer science, it is often imperative to get ideas on the size of data structures or some set of information. When counting a set  $A$ , we will usually do the following:

1. Write a procedure that will generate each item in  $A$  such that we don't
  - Overcount: count the same thing in  $A$  twice or more (ie. non-injective)
  - Undercount: never count something in  $A$  (ie. non-surjective)
2. Extract the number of  $A$  from the procedure

To illustrate the concept, consider the following example:

**Example 1.** We begin by counting the number of elements in the set  $[10] \times [10]$ . This is the number of ordered pairs, each coordinate coming from the first ten positive integers.

We begin by writing a procedure that will generate each element in the set exactly once. We will do this using pseudo-code for now, but later will we even more lax syntax.

```
for (i=1; i <=10; i++) { // This chooses the first coordinate
  for (j=1; j<=10;i++) { // This choses the second coordinate
    say '(i,j)' // Reports having created the pair (i,j)
  }
}
```

It is very clear this pair generates never generates the same pair twice, and it of course generates each pair at least once. This says that our procedure is correct. From the procedure, it is very clear that there are  $10 \cdot 10 = 100$  such pairs, and each iteration of the outer loop yield ten different pairs.

For such a simple example, it is difficult to see exactly why we must go through such elaborate depths to count the size of a set. After all, many would say it's obvious without doing any work that the set above as 100 members.

But, the sets we are counting get enormously difficult. Imagine, for instance, if I asked you to count the number of possible dice rolls with three dice where no more than one die shows an odd number. Or, perhaps I ask you to count the number of 15 digit, decimal numbers that are neither even nor divisible by 7.

These are both more complicated sets, and it's tough to know where to begin counting without thinking of it procedurally.

### 1.1 Rule of Sums

We first encounter our first rule for counting. We will express it first in terms of sets, and then try to extract what we can heuristically out of it.

*Fact 1.* If  $A \cap B = \emptyset$  and  $\Omega = A \cup B$  then  $|\Omega| = |A| + |B|$

What does this mean, and more importantly how can it help us? Well, we have already talked about a partition of a set; that is some sets which are all subsets of the larger, which span the set (everything is in some part of the partition; ie it is surjective) and are pairwise disjoint (everything is in at most one part of the partition; ie it is injective).

If one partitions a set  $\Omega$ , then one can count the parts of the partition separately instead of together.

I said the above as a fact, but is actually a theorem of cardinality. Can you see the proof?

**Example 2.** Let us count the number of ways to role two identical dice so that exactly one of the dice is either a 3 or 6. Notice, we can do cases on whether that one dice is a 3 or 6, hence partitioning our universe.

If a 3 is rolled, there are 4 ways to do this task; namely the other die can be 1, 2, 4, 5.

Similarly, if 6 is rolled.

Therefore, the total number of ways the above can happen is 8 ways.

*Remark 1.* Procedurally the rule of sums amount into doing cases on a particular event. You count count the number of elements given that one event occurred (a 3 is rolled, for instance) and then the number of events that that even failed to occurred (a 3 is not rolled, which in our problem meant that a 6 had to be rolled).

*Remark 2.* No matter how simple or complex a counting argument is, you should always “debug it.” Counting arguments are always, in one form or another, code. And you debug code by trying to make it break. Try to run through your procedure and get it to generate the same thing twice. Try to find some output that seems hard for the procedure to generate. Look at the fringe cases, but also at the typical ones. Take a typical example and see how your program generates.

For instance, in the above a typical dice role as described would be 4 and 3. Did I count this case? Yes; it was counted in the case where 3 was rolled. I accounted for the fact that the other die could be a 4. Is it counted more than once? No, because given the output, (4 and 3) I know exactly where my procedure generated it, and it can be generated no other place.

## 1.2 Common Counts

What we do now is create some very common “subroutines” that we can use in our counting procedures. This will greatly reduce the amount of code that we need to generate ourselves.

### 1.2.1 Rule of Products

Let us suppose that we had  $k$  many events, and each of these events can be paired with  $n$  events. The situation that comes to mind may be that we have  $k$  shirts and  $n$  pants. An outfit consists of a shirt and pants, therefore how many outfits do we have?

**Theorem 1** (Rule of Products). *The number of ways to perform a two part task where there are  $k$  ways to perform the first part and  $n$  ways to perform the second is  $n \cdot k$ .*

*Proof.* We do cases on which of the first task is done. If it is the first of  $k$ , then there are  $n$  ways to do the next. Similar, if it is the second, then there are  $n$  ways to the the next. So on, for each  $1 \leq i \leq k$ . Therefore, there are

$$\underbrace{n + n + \cdots + n}_{k\text{-many}}$$

ways to complete the task, which is  $n \cdot k$  □

**Example 3.** How many 4-digit decimal pins can there be? We write it as a procedure:

**Step 1** Pick the first digit: 10 ways

**Step 2** Pick the second digit: 10 ways

**Step 3** Pick the third digit: 10 ways

**Step 4** Pick the fourth digit: 10 ways

Therefore, by iterating the rule of products several times, we get the answer is  $10 \cdot 10 \cdot 10 \cdot 10 = 10^4$

### 1.2.2 Permutations

**Example 4.** If you have  $n$  people, how many ways are there to arrange them in a line?

Well, we use the rule of products:

<b>Step 1</b>	Pick the first person for line:	$n$ ways
<b>Step 2</b>	Pick the second person for the line: <i>the person chosen for the first of the line cannot be chosen for the second</i>	$n - 1$ ways
<b>Step 3</b>	Pick the third person for the line:	$n - 2$ ways
	$\vdots$	
<b>Step <math>n - 1</math></b>	Pick the $n - 1$ th person for the line:	2 ways
<b>Step <math>n</math></b>	Pick the $n$ th person in line	1 way

This makes the total  $n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot 2 \cdot 1$

**Definition 1.** Because of the importance of the above subroutine, we give it a name. An arrangement of  $n$  objects in a line (where order matters, of course) is a **permutation** of  $n$  objects.

The example above gives us a proof that the number of permutations of  $n$  objects is  $n!$ , which is  $n$  **factorial**.

**Example 5.** If you have  $n$  people, how many ways are there to choose a leader and a vice leader?

We use the rule of products:

<b>Step 1</b>	Pick the leader	$n$ ways
<b>Step 2</b>	Pick the vice-leader	$n - 1$ ways

The rule of products makes the total  $n \cdot (n - 1)$ .

**Definition 2.** Notice the above situation is equivalent to lining up 2 people from the group of  $n$ , and then making the first person the leader and the second the vice leader.

In general, the number of ways to line up  $k$  people out of  $n$  is  $\frac{n!}{(n-k)!}$ . Notice, this is

$$\frac{n!}{(n-k)!} = n(n-1)(n-2)\cdots(n-k+1)$$

This is called a **falling factorial**. There are a lot of notations for this, but we will avoid using any since there is no standard notation, and  $\frac{n!}{(n-k)!}$  is not so hard to write.

### 1.2.3 Subtraction Principle

**Example 6.** Suppose that we want to count the number of 4 digit, decimal pins that have at least two digits.

This might be a daunting task, but instead of counting it directly, we can begin by counting the number of 4 digit pins. We have done this already:  $10^4$ . And now, we need only count the number of sequence which do not have at least two digits and subtract.

Well, every number that does not have two digits has all identical digits. Thus our procedure is simple:

<b>Step 1</b>	Choose Repeated Digit	10 ways
---------------	-----------------------	---------

Therefore, there are 10 ways to choose a pin without at least two digits. Therefore there are  $10^4 - 10$  ways to choose a 4 digit number where there are at least 2 digits.

*Fact 2.* The general principle is this: If we know  $|\Omega|$  and we want to count  $A$ , where  $A \subseteq \Omega$  then if we can count  $A^c = \Omega \setminus A$  then we can get  $A$ , by

$$|A| = |\Omega| - |A^c|$$

### 1.2.4 Division Principle

The division principle is extremely important and wonderfully powerful, but also very dangerous. Here is the principle:

*Fact 3.* If we are counting a set  $S$  and we can partition  $S$  into  $k$  parts all of equal size  $n$ , then we know that

$$k = \frac{|S|}{n}$$

When we do this procedure, we are actually counting the number of equivalence classes with respect to the partition (there are  $k$  equivalence classes, each with  $n$  members). Because of this, we call the above procedure as **modding out by** whatever the relation was.

*Remark 3.* It is very important, obviously, that we need to know the above is a partition. Otherwise, we might not even get an integer!

**Example 7.** Let us once again count the number of ways to arrange  $k$  people from a set of size  $n$ . Here, we begin by being very reckless and ordering all the people, of which there of course  $n!$  ways.

Now, from this, we read off the first  $k$  people in line as the people chosen, and the order as the order. The problem is, we overcounted a lot. For instance, if the people were numbers 1, 2, 3, 4 and  $k = 2$ , then notice we counted the arrangement (1, 2) twice:

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

We want to create an equivalence relation where we relate two strings if they are identical in the first  $k$  spots. Notice, for each of these there are  $(n - k)!$  things in each equivalence class; that is, for every fixed arrangement of the first  $k$  objects, there are  $(n - k)!$  arrangements of the remaining objects.

As we want to view those all as the same counting, we need to divide by  $(n - k)!$ ; that is, we mod out by the lines being equal for the first  $k$  spots. This gives  $\frac{n!}{(n-k)!}$  as the answer, as we had before.

### 1.2.5 Combinations

**Example 8.** Before we elected a leader, and a vice leader. What if, instead, in the spirit of a representative democracy we want to elect a congress of  $k$  people to represent the population of size  $n$ . How would we do this?

One could imagine that we would line up the  $n$  people, which there are  $n!$  ways. Being in the first group of  $k$  people would say that you are in the congress. But, of course, this overcounts as we'd like the order not to matter among the first  $k$  people, and among the last  $n - k$  people. Therefore, we define an equivalence relation on this set, relating the string if the first  $k$  people are identical. There are  $k!$  members of each equivalence class, and  $\frac{n!}{k!}$  equivalence classes.

Similarly, we define an equivalence relation on equivalence classes of this set, by relating two things if the last  $n - k$  elements of the same. There are  $(n - k)!$  such members of each equivalence class, making  $\frac{n!}{k!(n-k)!}$  many equivalence classes.

Thus, the answer is  $\frac{n!}{(n-k)!k!}$

**Definition 3.** This leads to the very important notation of a combination. A combination of size  $k$  from a set of a size  $n$  is a way to pick  $k$  elements from that set (the order doesn't matter). Because of how useful of an idea this is, we have a special notation:  $\binom{n}{k}$  is the number of combinations of  $k$  objects from a set of size  $n$ .

The above proves that the number of combinations from the a set of size  $n$  of size  $k$  is  $\frac{n!}{(n-k)!k!}$ .

*Remark 4.* It is important to note that  $n!$  is not defined to be the number of permutations of size  $n$ , but that is a theorem that one proves. The factorial is defined to be the product of the first  $n$  numbers.

On the other hand,  $\binom{n}{k}$  is defined to be the number of way to chose  $k$  objects from a set of size  $n$ . One proves that (like we did above), and shows that this is actually equal to  $\frac{n!}{(n-k)!k!}$ . So, the number of combinations is not defined to be this formula.

### 1.3 Examples

**Example 9.** How many orderings are there of a deck of 52 cards such that all the suits of the same suit are ordered together

<b>Step 1</b>	Arrange Clubs	13! ways
<b>Step 2</b>	Arrange Spades	13! ways
<b>Step 3</b>	Arrange Diamonds	13! ways
<b>Step 4</b>	Arrange Hearts	13! ways
<b>Step 5</b>	Arrange the four piles	4! ways

This makes the total  $4!(13!)^4$

**Example 10.** How many distinct positive divisors does  $10^{100}$  have?

This number is  $2^{100} \cdot 5^{100}$ . Therefore, for power of 2 less than or equal to one hundred and any power of 5 less than or equal to one hundred, if we take their product, it divides  $10^{100}$ . So we are really counting pairs from [101] (0 is possible), and we get our factor by taking the first thing in pair and putting it as a power of 2 and the second as a power of 5.

Thus, the answer is  $101^2$ .

**Example 11.** Count the number of poker hands of each type.

Pairs:

<b>Step 1</b>	Choose rank of the cards	13 ways
<b>Step 2</b>	Choose suit of that rank	$\binom{4}{2}$ ways
<b>Step 3</b>	Choose the rest of the cards	$\binom{49}{3}$ ways

*Note: the two cards as the same rank as the ones we chose are disqualified as that would make more than a pair*

This makes the total  $13\binom{4}{2}\binom{49}{3}$

Two Pair:

<b>Step 1</b>	Choose two different ranks	$\binom{13}{2}$ ways
<b>Step 2</b>	Choose suits for higher rank chosen	$\binom{4}{2}$ ways
<b>Step 3</b>	Choose suits for lower rank chosen	$\binom{4}{2}$ ways
<b>Step 4</b>	Choose the other card	44 ways

This makes the total  $44\binom{13}{2}\binom{4}{2}^2$

Three of a Kind (exercise)

Straight

<b>Step 1</b>	Choose lowest card in straight	10 ways
<b>Step 2</b>	Choose suit for each cards	$4^5$ ways

This makes the total  $10 \cdot 4^5$

Flush

<b>Step 1</b>	Choose suit	4 ways
<b>Step 2</b>	Choose ranks	$\binom{13}{5}$

This makes the total  $4\binom{13}{5}$

Full House (exercise)

Striaight Flush (exercise)

**Example 12.** How many ways can 6 indistinguishable rooks be placed on a  $6 \times 6$  chessboard so none can attack each other? We begin by noticing that each rook needs to be in it's own row.

<b>Step 1</b>	Pick the place for the rook in the first row	6 ways
<b>Step 2</b>	Pick the place for the rook in the second row	5 ways
	$\vdots$	$\vdots$
<b>Step 6</b>	Pick the place for the rook in the last row	1 way

So the total is  $6!$  ways.