Lecture 2: Basic counting

Rajat Mittal*

IIT Kanpur

There are various instances in mathematics when we are interested in the size of a special set, a set which has specified properties. Sometimes, we are just happy to show the existence of such a set. Let us take a look at some examples.

- How many numbers are there in $2000, 2001, \cdots, 5678$ which are divisible by both 3 and 8.
- Does there exist a strategy for the first player/second player to ensure a draw in the game of tic-tac-toe.
- How many different phone numbers are possible with 8 digits which do not start with 9.
- What is the minimum number of ties possible if all students selected in JEE have scored marks between 100 and 250.
- In how many ways can we distribute n letters in n envelops, so that no letter goes in the correct envelope.

All these problems are called *counting problems*. Counting problems arise in almost every aspect of computer science. In this lecture we will learn some basic techniques and principles for counting.

1 Basic counting

There are two very simple rules used extensively to count.

1. *Sum rule:* We need to choose one element from two sets, first containing a elements and the other b. If the two sets are *disjoint* then it can be done in $a + b$ ways.
2. *Product rule:* We need to choose two elements, one *each* from two sets, first containing a elements and the other b. Then, the total number of ways is ab.

We have taken only two sets, but the rule can be generalized to multiple sets easily. Let us do some examples.

Example 1. How many bit strings are there of length 7?

There are 7 positions. In each of these positions, there are two possibilities. Applying product rule, there are $2^7 = 128$ possibilities.

Exercise 1. How many numbers are there between 1000 and 9999 (including both) which are divisible by 3?

Count in the range $\{1, \ldots, 1000\}$ and $\{1, \ldots, 9999\}$.

Example 2. How many palindrome words are there of length n?
If n is even then $26^{\frac{n}{2}}$. If n is odd then $26^{\frac{n+1}{2}}$.

Example 3. How many ways are there to put m balls into n bins.

1. Balls are distinct and bins are distinct:
   Every ball has n choices. So, n^m.

   Exercise 2. Why is the answer not m^n by looking at the opposite argument?

* Thanks to Nitin Saxena for his notes from the previous iteration of the course.

2. Balls are identical but bins are distinct:
   This corresponds to ordered partitions of the number m (in $\leq n$ parts). So, consider a string of m zeroes and $n-1$ ones, and permute them. Every permutation corresponds to a way of placing m balls in n bins. Here, ones signify the division between the bins. In other words, i-th bin contains the number of zeroes between the $(i-1)$-th and the i-th one.
   The count is
   $$\frac{(m+n-1)!}{m! \cdot (n-1)!} = \binom{m+n-1}{m}.$$
3. Balls are distinct but bins are not:
   This corresponds to (unordered) partitions of the set $\{1, \ldots, m\}$ (in $\leq n$ parts). This is a difficult counting problem. We will cover it under *Bell's number* later.
4. Balls (respectively bins) are identical:
   This corresponds to the (unordered) partitions of the number m (in $\leq n$ parts). Again, this is a difficult counting problem. We will cover it under *generating functions* later.

Generally, we need to use both rules, sum and product, simultaneously to solve counting problems. Let us take another example.

Example 4. Suppose, a password in the internal computer system needs to be 8 to 10 characters long. It can use uppercase/lowercase letters or digits, but need to have at least one digit. How many such passwords are there?

By sum rule, we need to sum the number of passwords of length 8,9 and 10. Let us count the total number of passwords of length 8. There are 52 choices of letters and 10 choices for digits. So there are 62^8 passwords, but there are 52^8 passwords which do not contain any digit. So, total number of valid passwords of length 8 are $62^8 - 52^8$.

So, the total number of passwords are $62^8 + 62^9 + 62^{10} - 52^8 - 52^9 - 52^{10}$.

1.1 Counting in two ways

Let us consider the following problem. Show that in a conference, the number of members who shake hands an odd number of times is even.

We will count the number P of ordered pairs (m_i, m_j), where member m_i shook hands with another member m_j. We know that P is twice the number of total handshakes. Hence, P is even.

Counting P another way, it is the sum of handshakes done by each member. If the number of members who shook hands an odd number of time is odd then P will be odd (why?). But, since P is even, it implies that the number of members who shook hands an odd number of times is even.

This trick is called *counting in two ways*. The idea is to count one particular quantity in two different ways. Since we know that any counting should give identical results, we can derive certain properties. Let us go through a few examples.

Example 5. According to DOAA office, a 1-st year student takes exactly 6 courses in his first semester. The DOAA office notified that the enrollment in first semester courses was $34, 56, 82, 45, 29, 63, 92$. Show that there is an error in the data provided by the DOAA office. Assume that these course are only offered for first year students.

We will count the pairs (x, y) where x is a student taking course y. Suppose there are s students, the number of pairs is $6s$ (counting student-wise). Counting course-wise, total number of pairs is $34 + 56 + 82 + 45 + 29 + 63 + 92 = 401$. By double counting, $6s = 401$. Since 401 is not divisible by 6, there is an error in enrollment data.

Exercise 3. According to DOAA office, a 1-st year student is supposed to take 6 courses in his first semester. If he/she is allowed reduced course load, then they have to take 4 courses. The DOAA office notified that the enrollment in first semester courses was $34, 56, 82, 45, 29, 63, 92$. Show that there is an error in the data provided by the DOAA office. Assume that these course are only offered for first year students.

Suppose there are s students with full load and t students with reduced load, pairs is $6s + 4t$. By double counting, $6s + 4t = 401$. Since $6s + 4t$ is divisible by 2 but 401 is not, there is an error.

Example 6 (Fermat's little theorem). Fermat is famous for his last theorem which was proved recently by Andrew Wiles. His another famous contribution is the little theorem of Fermat, which states that $a^p - a$ is divisible by p for all positive integers a. We will see multiple algebraic proofs and extensions of this theorem later. Let us prove it using counting today. Before going through the solution, give it a try first.

Exercise 4. Read about Fermat's last theorem.

Suppose we want to create a word of length p where every letter can take a choices (the alphabet size is a). The total number of words possible is a^p by product rule. On the other hand, every word can be shifted cyclically p times, creating *orbits* of these p elements. It is easy to see that these orbits are distinct. If x is in the orbit of y, then y is in the orbit of x.

Exercise 5. This will show that p divides a^p, what is the problem?

To make the above proof work, we need to show that these orbits are disjoint and contain exactly p elements. Observe that if all letters are same in a word then there is only one element in its orbit. There are exactly a such words.

Exercise 6. Show that the orbits in the remaining $a^p - a$ words are disjoint and contain exactly p elements.

Use that p is a prime, so the sequence can't be a perfect repetition of a smaller sequence.

There are $a^p - a$ words removing those which have the same letter repeated p times. If there are t orbits then there are pt words. By double counting, $a^p - a = pt$ and hence p divides $a^p - a$.

2 Binomial coefficients

First, let us talk about something you have already seen before, permutations and combinations. Say, we are interested in collecting items from a given set. There are two possibilities, we might be interested in ordered or unordered collection. Fix a set S and say that it has n elements.

The first one, ordered arrangement of a set, is called a *permutation* of the set. A r-permutation is an ordered set of r elements from the set.

Exercise 7. How many r-permutations of S are there? $\frac{!(r-n)}{!n}$

Similarly total number of ways of selecting an r size subset (unordered) from S is,

$$\binom{n}{r} := \frac{n!}{r!(n-r)!}.$$

The quantities $\binom{n}{r}$ are called *binomial coefficients*. We will assume that you are familiar with these basic notions in permutation and combination. If not, please look at any introductory literature on permutation and combination.

2.1 Properties of binomial coefficients

Binomial coefficients are very important in discrete mathematics and have many nice properties. Let us take a look at few of them. Remember that,

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}.$$

Exercise 8. Prove that $\binom{n}{r} = \binom{n}{n-r}$.

It is an easy exercise by looking at the formula. Though, we can see it combinatorially too. $\binom{n}{r}$ is the number of ways of selecting r objects from a set of n objects. But, selecting r objects is same as discarding $n-r$ objects, and hence selecting $n-r$ objects. So, we get the equality.

Let us prove another theorem about the sum of binomial coefficients.

Theorem 1. *For any positive integer n,*

$$\sum_{r=0}^{n} \binom{n}{r} = 2^n.$$

Proof. We will prove the theorem using double counting. We want to count, how many committees can be formed from a set of n people? Notice that there is no restriction on the number of people in the committee.

One way, every person has two choices, *to be or not to be* in the committee. By product rule, total number of committees is 2^n.

Another way, a committee can have r people, with r ranging from 0 to n. By sum rule, total number of committee is $\sum_{r=0}^{n} \binom{n}{r}$.

Since both these ways should give the same count,

$$\sum_{r=0}^{n} \binom{n}{r} = 2^n.$$

□

The next identity we discuss is called *Vandermonde's identity.*

Theorem 2 (Vandermonde's identity). *For any two positive integers m, n and a positive integer $k \leq m, n$;*

$$\sum_{r=0}^{k} \binom{n}{r}\binom{m}{k-r} = \binom{m+n}{k}.$$

Proof. We will prove the theorem using double counting (combinatorial argument). We want to count, how many teams of k players can be formed from n players in India and m players in Pakistan?

One way, there are $m+n$ players and k players need to be selected. By definition, this is $\binom{m+n}{k}$.

Another way, we can choose r players from India, then $k-r$ players need to be selected from Pakistan. By sum rule, total number of teams is $\sum_{r=0}^{k} \binom{n}{r}\binom{m}{k-r}$.

Since both these ways should give the same count,

$$\sum_{r=0}^{k} \binom{n}{r}\binom{m}{k-r} = \binom{m+n}{k}.$$

□

To compute $\binom{n}{k}$, one way is to compute using the formula. Sometimes, $\binom{n}{k}$ could be small but $n!$ could be big, leading to problems with this idea. Another way is to calculate the binomial coefficients using their smaller counterparts.

Exercise 9. Using the formula of binomial coefficients, prove that,

$$\binom{n}{r} = \binom{n-1}{r} + \binom{n-1}{r-1}.$$

This is called a recurrence relation. It gives us a recursive way to compute $\binom{n}{k}$.

Note 1. Trivially applying this idea will get us into trouble. We need to use the ideas of dynamic programming to calculate this quantity efficiently.

Exercise 10. Read about Pascal's triangle. The identity above (Ex. 9) is known as Pascal's identity.

We will study recurrence relations next.

3 Recurrence relations and generating functions

Recursion is a helpful tool to solve problems in computer science and mathematics. We will look at recursion as an aid to counting. We already saw a recursive relation for $\binom{n}{k}$ (Ex. 9). It was easy to see the explicit formula for the binomial coefficients. Still, computing it was made easier by the recursive relation. In some cases, even the explicit formula is not easy to find and the recursive formula helps us find one. Let us take an example.

Suppose rabbit population needs to be introduced to an island. A pair of rabbits does not breed in its first month, and produces a pair of offspring in each subsequent month (assume that there are no deaths). Starting with one newborn pair, what will be the number of pairs after n months ?

Say, we denote the number of pairs in the n^{th} month by F_n ($F_0 := 1, F_1 := 1$). There will be two kinds of rabbit-pairs making up F_n: new-born (≤ 1 month old) and older. F_{n-2} will be the new-born pairs, while F_{n-1} would be the older ones. So,

$$F_n = F_{n-1} + F_{n-2}\,.$$

This is called a recurrence relation for F_n. We gave a *combinatorial argument* for its proof. With the initial condition ($F_0 = F_1 = 1$), this recurrence gives us an easy algorithmic way to compute the population in the n-th month.

Many a times, it is hard to come up with an explicit formula for a mathematical quantity, but recurrence relation gives us valuable information about it.

The above sequence is very special and is known as *Fibonacci sequence*. That is, the numbers F_n are called Fibonacci if they satisfy the recurrence $F_n = F_{n-1} + F_{n-2}$ with the initial condition $F_0 = F_1 = 1$.

3.1 Generating functions

To find a "closed form" expression from the recurrence of F_n, we form a *formal power series* and studies it.

The power-series $\phi(t) = \sum_{i\geq 0} F_i t^i$ is called the *generating function* for sequence F_i.

Exercise 11. Why is it not called a polynomial but a power-series?

The expression is bounded on the left-end but unbounded on the right-end.

We will be doing additions, multiplications and other operations on these power-series without worrying about the notion of convergence (or avoid substituting any real value to t). These are known as *formal power series*. The justification for not worrying about the convergence is outside the scope of this course.

What can we do with the generating function? Say, we continue studying the Fibonacci sequence. If we multiply it with t, we get,

$$t\phi(t) = \sum_{i\geq 0} F_i t^{i+1} = \sum_{i\geq 1} F_{i-1} t^i\,.$$

Notice how we changed the summation index. Convince yourself that it works. Multiplying it again by t,

$$t^2\phi(t) = \sum_{i\geq 0} F_i t^{i+2} = \sum_{i\geq 2} F_{i-2} t^i\,.$$

Since $F_{i-1} + F_{i-2} = F_i$ (for $i \geq 2$), adding the above two equations will give us the original generating function. Almost (why?) !!

$$\phi(t) - 1 = t\phi(t) + t^2\phi(t)\,.$$

All the coefficients for terms higher than t^2 agree. The constant term 1 appears by comparing the constant coefficient and the coefficient of t. This is where we use initial conditions !! We get the formula for $\phi(t)$,

$$\phi(t) = \frac{1}{1 - t - t^2}.$$

We can factorize the polynomial, $1 - t - t^2 = (1 - \alpha t)(1 - \beta t)$, where $\alpha, \beta \in \mathbb{R}$.

Exercise 12. Show that $\alpha = \frac{1+\sqrt{5}}{2}, \beta = \frac{1-\sqrt{5}}{2}$.

We can simplify the formula a bit more,

$$\phi(t) = \frac{1}{1-t-t^2} = \frac{c_1}{1-\alpha t} + \frac{c_2}{1-\beta t} = c_1(1+\alpha t+\alpha^2 t^2+\cdots) + c_2(1+\beta t+\beta^2 t^2+\cdots)\,.$$

We got the second equality by putting $c_1 = \frac{1}{\sqrt{5}}\alpha$ and $c_2 = -\frac{1}{\sqrt{5}}\beta$. The final expression gives us an explicit formula for the Fibonacci sequence,

$$F_n = \frac{1}{\sqrt{5}}(\alpha^{n+1} - \beta^{n+1})\,.$$

This is actually a very strong method and can be used for solving *linear recurrences* of the kind, $S_n = a_1 S_{n-1} + \cdots + a_k S_{n-k}$. Here $k, a_1, a_2, \cdots, a_k$ are constants. Suppose $\phi(t)$ is the recurrence for S_n, then by the above method,

$$\phi(t) = \frac{b_1 + b_2 t + \cdots + b_k t^{k-1}}{1 - a_1 t - a_2 t^2 - \cdots - a_k t^k} = \frac{c_1}{1-\alpha_1 t} + \cdots + \frac{c_k}{1-\alpha_k t}\,.$$

Where $\alpha_1, \cdots, \alpha_k$ are the roots (assumed distinct) of polynomial $x^k - a_1 x^{k-1} - \cdots - a_k = 0$. This is known as the *characteristic polynomial* of the recurrence.

The characteristic polynomial is obtained by replacing t with $\frac{1}{x}$ in the denominator

$$1 - a_1 t - a_2 t^2 - \cdots - a_k t^k.$$

This transformation ensures that $\alpha_1, \cdots, \alpha_k$ are roots of the characteristic polynomial. Notice that the characteristic polynomial depends only on the recurrence and not on initial condition.

Exercise 13. How are the coefficients $b_1, \cdots, b_k$ or $c_1, \cdots, c_k$ determined?

Initial conditions.

So we get $S_n = c_1\alpha_1^n + \cdots + c_k\alpha_k^n$.

For a linear recurrence if F_n and G_n are solutions then their linear combinations $aF_n + bG_n$ are also solutions. For a k term recurrence $S_n = a_1 S_{n-1} + \cdots + a_k S_{n-k}$, let $\alpha_1, \cdots, \alpha_k$ be the roots of the characteristic polynomial.

The possible solutions are $\alpha_1^n, \cdots, \alpha_k^n$ and their linear combinations,

$$S_n = c_1\alpha_1^n + \cdots + c_k\alpha_k^n.$$

The coefficients of the linear combination are fixed by the initial conditions.

Exercise 14. Does every polynomial over $\mathbb{C}$ has a complex root?

Fundamental theorem of algebra.

What happens when the characteristic polynomial has repeated roots? We can use the power series for $(1-\alpha t)^{-d}$. In this case the corresponding d possible solutions will be $\alpha_t^n, n\alpha_t^n, \cdots, n^{d-1}\alpha_t^n$. For example, if the roots are 1,2,3,3 for the characteristic polynomial of S_n, the function will look like,

$$S_n = c_1 + c_2 2^n + c_3 3^n + c_4 n 3^n.$$

Here, the coefficients c_1, c_2, c_3, c_4 will be determined by the initial conditions.

We would like to emphasize that you should focus on the techniques of generating function instead of learning formulas for very special kind of recurrence relations.

4 Exponential generating function

What do we do when the recurrence is *non-linear* ? We will now see some related methods.

A permutation is called an *involution* if all cycles in the permutation are of length 1 or 2 [1]. We are interested in counting the total number of involutions of $\{1, 2, \cdots, n\}$, call that $I(n)$. If we look at the base cases; $I(0) = I(1) = 1$, $I(2) = 2$ and $I(3) = 4$.

There can be two cases.

1. The number n maps to itself. This case will give rise to $I(n-1)$ involutions.
2. The number n maps to another number i. There are $n-1$ choices of i and then we can pick an involution for remaining $n-2$ numbers in $I(n-2)$ ways.

By this argument, we get a simple recurrence,

$$I(n) = I(n-1) + (n-1)I(n-2)\,.$$

Exercise 15. Why is this not a linear recurrence?

Since the coefficient of $I(n-2)$ is not a constant, we cannot apply the usual approach of generating functions. Even without getting an explicit formula for $I(n)$, the recurrence can give us some information about the quantity.

Theorem 3. *For $n \geq 2$, the number $I(n)$ is even and greater than $\sqrt{n!}$.*

Proof. Both the statements can be proven using induction.

Exercise 16. What will be the base case and the induction step ?

Check that $I(2), I(3)$ are both even. Verify that $1 + \sqrt{n-1} \geq \sqrt{n}$, for $n \geq 1$.

□

In the case of involutions, the regular generating function will not be of much help. We define *exponential generating function* for the sequence $I(n)$ to be,

$$\theta(t) := \sum_{k\geq 0} \frac{I(k)t^k}{k!}\,.$$

Exercise 17. Why is it called an *exponential* generating function?

Put $I(k) = 1$, for all k, and we get the series for the exponential function e^t.

Note 2. It is merely the generating function of $I(k)/k!$.

We can actually come up with a closed form solution for the exponential generating function of involutions. We will differentiate the function $\theta(t)$,

$$\begin{aligned}
\frac{d}{dt}\theta(t) &= \sum_{k\geq 1} \frac{I(k)t^{k-1}}{(k-1)!} && \text{(formal differentiation)} && (1)\\
&= \sum_{k\geq 1} \frac{I(k-1)t^{k-1}}{(k-1)!} + \sum_{k\geq 1} \frac{(k-1)\cdot I(k-2)\cdot t^{k-1}}{(k-1)!} && \text{(recurrence relation)} && (2)\\
&= \theta(t) + t\sum_{k\geq 2} \frac{I(k-2)t^{k-2}}{(k-2)!} && \text{(second term's first entry is zero)} && (3)\\
&= \theta(t) + t\theta(t)\,. && && (4)
\end{aligned}$$

[1] Every permutation can be decomposed into cycles. For example, (12)(3)(45)(67)(8) is an involution. Permutation (123) is not an involution.

This transforms into the differential equation,

$$\frac{d}{dt}\log\theta(t) = 1 + t\,.$$

We can solve this, as,

$$\theta(t) = e^{t+\frac{t^2}{2}+c}\,.$$

Comparing the constant coefficient, we get $c = 0$ (since $I(0) = 1$). Thus,

$$\theta(t) = e^{t+\frac{t^2}{2}}\,.$$

Note 3. Again we should notice that the power series we are considering are not shown to be well behaved (convergence etc.). But there is a justification for being able to differentiate and do other *formal* operations on them; the details are outside the scope of this course.

Exercise 18. The number of partitions of a set with n elements is called the *Bell number*, B_n. Show that it satisfies the recurrence,

$$B_n = \sum_{i=1}^{n} \binom{n-1}{i-1} B_{n-i}\,.$$

This is equal to the number of equivalence relations on a set with n elements.
Also, this is the case of putting n distinct balls into n identical bins.

$$\frac{B_n}{(n-1)!} = \sum_{i=1}^{n} \frac{1}{(i-1)!} \cdot \frac{B_{n-i}}{(n-i)!}\,.$$

Define $B_0 := 1$. Let the n elements be $[n]$. In a partition, let $i \in [n]$ be the size of the part that contains n. The number of such partitions is $\binom{n-1}{i-1} \cdot B_{n-i}$. The recurrence obtained is a good case for using exponential generating function:

References

1. K. H. Rosen. Discrete Mathematics and Its Applications. *McGraw-Hill*, 1999.
2. N. L. Biggs. Discrete Mathematics. *Oxford University Press*, 2003.