

zscoder's blog

[Tutorial] Generating Functions in Competitive Programming (Part 1)

By [zscoder](#), [history](#), 5 years ago, 

Hi everyone! Inspired by the recent [Codeforces Round 641](#), I decided to write an introductory tutorial on generating functions here. I am by no means an expert in generating functions so I will write about what I currently know about them. [jqdai0815](#) has written a really interesting [blog](#) here on Codeforces about more advanced applications of generating functions, but I think there is no English tutorial on the basics of this topic yet (or at least on CP sites). Thus, I would like to share about this topic here.

I plan to split this tutorial into two parts. The first part (this post) will be an introduction to generating functions for those who have never learned about them at all, and some standard examples and showcases of generating functions. The second part will be a collection of several applications of generating functions in CP-style problems. If you are already familiar with generating functions, you may just skip to [Part 2](#) directly.

In this part, many examples are from the book [generatingfunctionology](#) which I think is also a great introductory material to generating functions.

What are Generating Functions?

Let's say we have a sequence a_0, a_1, a_2, \dots . We associate with a a series A which "encodes" the terms in a with its coefficients.

Formally, for a sequence of numbers $\{a_i\}_{i=0}^{\infty}$, we define the **ordinary generating function (OGF)** of a to be $A(x) = \sum_{i=0}^{\infty} a_i x^i$.

For example, consider the Fibonacci sequence f with the terms $0, 1, 1, 2, 3, 5, 8, \dots$. Then,

$$F(x) = 0 + x + x^2 + 2x^3 + 3x^4 + 5x^5 + 8x^6 + \dots$$

You may imagine that you are putting the (infinitely many) terms of the sequence in a line, and assigning a power of x to each term of the sequence in order. Adding them up, you get an "infinite polynomial" which somewhat encodes the sequence. The nice thing about generating functions is that sometimes the series is nicer to play around with which will sometimes uncover surprising properties of the sequence.

There are other types of generating functions, such as the Exponential Generating Function (EGF) and Dirichlet Generating Function. We will look at some examples of EGF later in this post, but for the next few examples we will focus on OGFs.

Before that, we introduce a simple notation. For a series $A(x) = \sum_{n \geq 0} a_n x^n$, we let $[x^n]A(x) = a_n$ (i.e. the coefficient of x^n in A).

Simple Examples of OGFs

Let's start with a very simple example. What is the OGF of the sequence $1, 1, 1, \dots, 1$? By definition, we have

$$A(x) = 1 + x + x^2 + x^3 + \dots$$
 Does this series look familiar?

$A(x)$ is actually a geometric series with common ratio x ! Thus, we can use the geometric series formula to write $A(x) = \frac{1}{1-x}$.

Note: Don't we need to care about convergence issues like $|x| < 1$? Well, it depends on what you want to do with your series. We will work on [formal power series](#) most of the time, which allows us to ignore the issue of convergence. However, this also means that we can't "substitute" x as some fixed value without care. For example, when $x = 2$, the series $A(x)$ diverges but for say $x = -\frac{1}{2}$, $A(x)$ converges. In this post, we won't deal with the analytic properties of the series most of the time. If you really need to substitute values though, the general rule of thumb is that you can do it if the series converges.

We can manipulate generating functions very much like how we would manipulate other algebraic expressions. Here is a classic example.

Example. Consider the sequence f_n defined by $f_0 = 0$, $f_1 = 1$ and $f_n = f_{n-1} + f_{n-2}$ for $n \geq 2$. Find the OGF of f (we usually denote it with a capital letter, say F).

Clearly, f_n is the n -th Fibonacci number. We will use the recurrence relation to find the OGF of f_n .

Firstly, we need to make the terms of the series appear. The easiest way to do this is to multiply the recurrence relation by x^n to obtain $f_n x^n = f_{n-1} x^n + f_{n-2} x^n$.

Next, we sum up the terms on both sides over all valid n (in this case $n \geq 2$) to obtain:

$$\sum_{n=2}^{\infty} f_n x^n = x \sum_{n=2}^{\infty} f_{n-1} x^{n-1} + x^2 \sum_{n=2}^{\infty} f_{n-2} x^{n-2}.$$

This is equivalent to:

$$F(x) - f_0 x^0 - f_1 x^1 = x(F(x) - f_0 x^0) + x^2 F(x).$$

$$\Rightarrow F(x) - x = (x + x^2)F(x)$$

$$\Rightarrow F(x)(1 - x - x^2) = x$$

$$\Rightarrow F(x) = \frac{x}{1 - x - x^2}$$

Thus, we obtain the OGF of Fibonacci numbers.

Let's see another example of OGF of a common sequence.

Example. The Catalan numbers c_n are defined by $c_0 = 1$ and $c_{n+1} = \sum_{i=0}^n c_i c_{n-i}$ for $n \geq 0$. Find the OGF of c_n .

Again, our strategy is to multiply a power of x to both sides and summing up for all n . We obtain:

$$\sum_{n=0}^{\infty} c_{n+1} x^{n+1} = \sum_{n=0}^{\infty} \sum_{i=0}^n c_i c_{n-i} x^{n+1} = x \sum_{n=0}^{\infty} \sum_{i=0}^n c_i x^i c_{n-i} x^{n-i}$$

The LHS is easy to interpret: it is just $C(x) - 1$.

How do we interpret the RHS? We claim that it is $x C(x)^2$. Consider the expansion of $C(x)^2$. Which terms contribute to the coefficient of x^n ? If we look at $C(x)^2 = (c_0 + c_1 x + c_2 x^2 + \dots)(c_0 + c_1 x + c_2 x^2 + \dots)$, we see that we can only obtain x^n by picking $c_i x^i$ from the first bracket and $c_{n-i} x^{n-i}$ from the second bracket. Hence, the coefficient of x^n in $C(x)^2$ is $\sum_{i=0}^n c_i c_{n-i}$, as desired.

Hence, we have $C(x) - 1 = x C(x)^2$, which is a quadratic equation in $C(x)$! Using the quadratic formula, we can obtain $C(x) = \frac{1 \pm \sqrt{1-4x}}{2x}$. Which sign should we choose? If we choose the + sign, then the numerator $\rightarrow 2$ as $x \rightarrow 0$, while the denominator $\rightarrow 0$, so the ratio will become infinite at 0. However, $C(x)$ can be expanded as a power series at $x = 0$, so $C(x)$ should converge at $x = 0$. Thus, we should choose the minus sign to obtain $C(x) = \frac{1 - \sqrt{1-4x}}{2x}$ (indeed, by L'Hopital Rule it converges to $c_0 = 1$ at $x = 0$).

Tip: Try looking for common sequences and see if you can derive the formula for their OGFs from scratch. It is really helpful if you can derive the intuition where you can **see** the functional equation directly from the recurrence.

OGFs in more than one variable

We don't have to limit ourselves to one variable. We can have multivariable OGFs. Let's look at the following simple example.

Example. The binomial coefficients $c(n, k)$ is defined by the recurrences $f(n, 0) = 1$ for $n \geq 0$, $f(0, n) = 0$ for $n \geq 1$ and $f(n, k) = f(n-1, k) + f(n-1, k-1)$ for $n, k \geq 1$. Find the OGF of $f(n, k)$.

We define the OGF $F(x, y) = \sum_{n \geq 0} \sum_{k \geq 0} f(n, k) x^n y^k$. As usual, we try to relate F with itself using the recurrence. We have

$$\sum_{n \geq 1} \sum_{k \geq 1} f(n, k) x^n y^k = x \sum_{n \geq 1} \sum_{k \geq 1} f(n-1, k) x^{n-1} y^k + xy \sum_{n \geq 1} \sum_{k \geq 1} f(n-1, k-1) x^{n-1} y^{k-1}.$$

Hence, we have

$$F(x, y) - \sum_{n \geq 0} x^n = x(F(x, y) - \sum_{n \geq 0} x^n) + xyF(x, y)$$

$$\Rightarrow F(x, y) - \frac{1}{1-x} = (x + xy)F(x, y) - \frac{x}{1-x}$$

$$\Rightarrow (1 - x - xy)F(x, y) = 1$$

$$\Rightarrow F(x, y) = \frac{1}{1-x-xy}$$

From the bivariate OGF, we can deduce some interesting identities in one-variable. For example, we have

$$F(x, y) = \frac{1}{1-x-xy} = \frac{1}{1-x(y+1)} = \sum_{k \geq 0} (y+1)^k x^k$$

Hence, $[x^n]F(x, y) = (y+1)^n$. However, $[x^n]F(x, y) = \sum_{k=0}^{\infty} f(n, k) y^k$, so $\sum_{k=0}^{\infty} f(n, k) y^k = (y+1)^n$. Note that this gives the same result as binomial theorem on $(y+1)^n = \binom{n}{0} y^0 + \binom{n}{1} y^1 + \dots + \binom{n}{n} y^n$.

It is more interesting to look at $[y^k]F(x, y)$ in terms of an OGF in x . We have

$$F(x, y) = \frac{1}{(1-x)-xy} = \frac{\frac{1}{1-x}}{1-\frac{x}{1-x}y} = \frac{1}{1-x} \left(1 + \frac{x}{1-x}y + \left(\frac{x}{1-x}\right)^2 y^2 + \dots \right)$$

Comparing coefficients, we obtain $[y^k]F(x, y) = \frac{x^k}{(1-x)^{k+1}}$, so using the same argument as before, we have

$$\sum_{n=0}^{\infty} f(n, k) x^n = \frac{x^k}{(1-x)^{k+1}}.$$

This identity is interesting because it allows us to “expand” $\frac{1}{(1-x)^{k+1}}$ in terms of the OGF of binomial coefficients! In particular, we have $[x^{n-k}] \frac{1}{(1-x)^{k+1}} = \binom{n}{k}$, so $[x^n] \frac{1}{(1-x)^{k+1}} = \binom{n+k-1}{k}$. This identity is very useful especially when dealing with sums involving binomial coefficients where k is fixed and n varies.

Exponential Generating Function

So far we have looked at ordinary generating functions. Now, let me introduce a new type of generating functions called the **exponential generating function**.

Definition: Let a_0, a_1, a_2, \dots be a sequence of numbers. Then, the EGF of a (say $A(x)$) is defined as $\sum_{i=0}^{\infty} \frac{a_i}{i!} x^i$.

In other words, the EGF is just the OGF but every term a_i is now divided by $i!$. Why the weird choice of division by $i!$? The next example will shed some light on this choice.

Example. Let b_n denote the n -th Bell number, which counts the number of ways to partition $\{1, 2, \dots, n\}$ into disjoint sets. For example, $b_3 = 5$, because there are 5 ways to partition $[3]$ into sets: 123; 12, 3; 13, 2; 1, 23; 1, 2, 3. Find the EGF of b_n .

Our first step is to look for a recurrence relation. Suppose you have this as a Div. 2 C problem. What is the simplest dp you can come up with?

We can fix the size of the set containing the element 1, say i . Then, there are $\binom{n-1}{i-1}$ ways to choose the other $i-1$ elements of the set, and b_{n-i} ways to partition the remaining $n-i$ elements. Hence, we obtain the simple recurrence formula

$$b_n = \sum_{i=1}^n \binom{n-1}{i-1} b_{n-i} = \sum_{i=0}^{n-1} \binom{n-1}{i} b_i \text{ for } n \geq 1.$$

By precomputing binomial coefficients, this is an $O(n^2)$ dp, which should be sufficient for a Div. 2 C. However, why stop here? Suppose the problem asks you to find b_n for $n \leq 3 \cdot 10^5$. Can you still solve this?

The answer is yes. Let's use our recurrence to find the EGF of b_n . Note that

$$b_n = \sum_{i=0}^{n-1} \binom{n-1}{i} b_i = \sum_{i=0}^{n-1} \frac{(n-1)!}{i!(n-1-i)!} b_i$$

$$\Rightarrow n \frac{b_n}{n!} = \sum_{i=0}^{n-1} \frac{b_i}{i!} \frac{1}{(n-1-i)!}$$

$$\Rightarrow \sum_{n \geq 1} n \frac{b_n}{n!} x^n = \sum_{n \geq 1} x \sum_{i=0}^{n-1} \frac{b_i x^i}{i!} \frac{x^{n-1-i}}{(n-1-i)!}$$

Now we see why EGFs are convenient for us. If our convolutions involve binomial coefficients (which is often the case when we deal with combinatorial objects), then multiplying EGFs kind of "automatically" helps us multiply our terms by a suitable binomial coefficient (more details later).

Back to our problem, we want to write everything in terms of $B(x)$. RHS is easy, since it is just $xB(x)e^x$ (Recall that $\sum_{n \geq 0} \frac{x^n}{n!}$ is the

Maclaurin series of e^x). However, the LHS needs a bit of work, since we have the unfortunate nb_n term instead of the b_n term. To deal with this obstacle, we use a common trick when dealing with formal power series. Let us **differentiate** $B(x)$, then multiply by x . Verify that if $A(x)$ is a formal power series with $A(x) = a_0 + a_1x^1 + a_2x^2 + \dots = \sum_{n \geq 0} a_n x^n$ then

$$xA'(x) = a_1x^1 + 2a_2x^2 + 3a_3x^3 + \dots = \sum_{n \geq 0} na_n x^n.$$

Thus, looking back at our equation, we have

$xB'(x) = xB(x)e^x$, which implies $\frac{B'(x)}{B(x)} = e^x$. If you are familiar with calculus, you will recognize that if we integrate both sides, we get $\ln B(x) = e^x + c$. Since $b_0 = 1$, $B(0) = 1$ and we have $c = -1$. Thus, $B(x) = e^{e^x-1}$ is our desired EGF.

So, how to find b_n in faster than $O(n^2)$ time. The idea is that we can find the first n terms of $e^{P(x)}$ in $O(n \log n)$ time, so we just need to compute the first few terms of our EGF and read off the answer! In this 2-part article, I will omit explaining how to do certain well-known polynomial operations in $O(n \log n)$ time or $O(n \log^2 n)$ time like $\sqrt{P(x)}$, $\ln(P(x))$ etc. There are already tutorials written for them (for example [cp-algorithms](#)). Hence, I will just quote that we can do those polynomial operations since that is not the main focus of this article.

Algebraic Manipulation of Generating Functions

Here are some common ways to manipulate generating functions and how they change the sequence they are representing. In this section, a_i , b_i will represent sequences and $A(x)$ and $B(x)$ are their corresponding generating functions (OGF or EGF depending on context which will be stated clearly). As an exercise, verify these statements.

Addition

For both OGF and EGF, $C(x) = A(x) + B(x)$ generates the sequence $c_n = a_n + b_n$.

Shifting

For OGF, $C(x) = x^k A(x)$ generates the sequence $c_n = a_{n-k}$ where $a_i = 0$ for $i < 0$. For EGF, you need to integrate the series $A(x)$ k times to get the same effect.

For OGF, $C(x) = \frac{A(x) - (a_0 + a_1 x + a_2 x^2 + \dots + a_{k-1} x^{k-1})}{x^k}$ generates the sequence $c_n = a_{n+k}$.

For EGF, $C(x) = A^{(k)}(x)$ generates the sequence $c_n = a_{n+k}$, where $A^{(k)}(x)$ denotes A differentiated k times.

Multiplication by n

For both OGF and EGF, $C(x) = xC'(x)$ generates the sequence $c_n = na_n$.

In general, you can get the new generating function when you multiply each term of the original sequence by a polynomial in n by iterating this operations (but I do not include the general form here to avoid confusion).

Convolution

This is really the most important operation on generating functions.

For OGF, $C(x) = A(x)B(x)$ generates the sequence $c_n = \sum_{k=0}^n a_k b_{n-k}$.

For EGF, $C(x) = A(x)B(x)$ generates the sequence $c_n = \sum_{k=0}^n \binom{n}{k} a_k b_{n-k}$ (verify this!). This is also why EGF is useful in dealing with recurrences involving binomial coefficients or factorials.

Power of Generating Function

This is just a direct consequence of convolution, but I include it here because it is so commonly used.

For OGF, $C(x) = A(x)^k$ generates the sequence $c_n = \sum_{i_1+i_2+\dots+i_k=n} a_{i_1} a_{i_2} \dots a_{i_k}$

For EGF, $C(x) = A(x)^k$ generates the sequence $c_n = \sum_{i_1+i_2+\dots+i_k=n} \frac{n!}{i_1! i_2! \dots i_k!} a_{i_1} a_{i_2} \dots a_{i_k}$

Prefix Sum Trick

This only works for OGF, but is useful to know. Suppose want to generate the sequence $c_n = a_0 + a_1 + \dots + a_n$. Then, we can take $C(x) = \frac{1}{1-x} A(x)$.

Why does this work? If we expand the RHS, we get $(1 + x + x^2 + \dots)A(x)$. To obtain the coefficient of x^n which is c_n , we need to choose x^i from the first bracket and $a_{n-i}x^{n-i}$ from $A(x)$, so summing over all i gives us $c_n = \sum_{i=0}^n a_i$.

List of Common Series

Before we delve into applications, I want to compile a short list of series that we will use frequently below. They are

$$\frac{1}{1-x} = 1 + x + x^2 + \dots = \sum_{n \geq 0} x^n$$

$$-\ln(1-x) = x + \frac{x^2}{2} + \frac{x^3}{3} + \dots = \sum_{n \geq 1} \frac{x^n}{n}$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots = \sum_{n \geq 0} \frac{x^n}{n!}$$

$$(1-x)^{-k} = \binom{k-1}{0}x^0 + \binom{k}{1}x^1 + \binom{k+1}{2}x^2 + \dots = \sum_n \binom{n+k-1}{n}x^n$$

Our goal in many problems will be to reduce the EGF or OGF involved in the problem into some composition of functions that we know above.

You can find a more complete list on Page 57 on [generatingfunctionology](#).

Generating Functions in Counting Problems

Generating functions is a powerful tool in enumerative combinatorics. There are so many applications that I can only cover a small fraction of them here. If you are interested in more examples of counting using generating functions, you can try the books [generatingfunctionology](#) and [Enumerative Combinatorics](#).

Here, I will show some classical examples of counting problems involving generating functions. In the next post, I will focus on CP problems which utilizes generating functions.

Catalan Numbers, revisited

We have shown before that the OGF of the Catalan numbers is $C(x) = \frac{1 - \sqrt{1-4x}}{2x}$. Suppose we want to find a closed-form formula for c_n . Of course, it is well-known that $c_n = \frac{1}{n+1} \binom{2n}{n}$, but let's pretend we don't know that yet. We want to "expand" our generating function $C(x)$, but there is a troublesome square root in our way.

This is where the **generalized binomial theorem** comes to our rescue. Before that, we need to define generalized binomial coefficients.

Definition. Let r be any complex number and n be a nonnegative integer. Then, $\binom{r}{n} = \frac{r(r-1)\dots(r-(n-1))}{n!}$.

This is the same as the usual binomial coefficients, but now we no longer require the first term to be a nonnegative integer.

Next, we show a special case of the theorem.

Theorem. Let r be a real number and n be a nonnegative integer, then

$$(1+x)^r = \sum_{n \geq 0} \binom{r}{n} x^n.$$

The proof is just differentiating the left side n times and compare the constant term. I leave this as an exercise.

$$\text{In particular, our mysterious function } \sqrt{1-4x} = (1-4x)^{\frac{1}{2}} = \sum_{n \geq 0} \binom{\frac{1}{2}}{n} (-4x)^n$$

$$= \sum_{n \geq 0} \frac{1}{2} \cdot \frac{-1}{2} \cdot \frac{-3}{2} \cdots \frac{-(2n-3)}{2} \cdot \frac{1}{n!} \cdot (-4)^n x^n$$

$$= 1 + \sum_{n \geq 1} \frac{(-1)^{n-1} (1 \cdot 3 \cdots (2n-3))}{2^n} \cdot \frac{(-4)^n}{n!} x^n$$

$$= 1 + \sum_{n \geq 1} -2^n \cdot \frac{(2n-2)!}{2^{n-1}(n-1)!} \cdot \frac{1}{n!} x^n$$

$$= 1 + \sum_{n \geq 1} \frac{-2 \cdot (2n-2)!}{(n-1)!n!} x^n$$

$$= 1 + \sum_{n \geq 1} -\frac{2}{n} \cdot \binom{2n-2}{n-1} x^n.$$

$$\text{Hence, } C(x) = \frac{1-\sqrt{1-4x}}{2x} = \frac{1}{2x} \left[1 - 1 - \sum_{n \geq 1} -\frac{2}{n} \cdot \binom{2n-2}{n-1} x^n \right]$$

$$= \sum_{n \geq 1} \frac{1}{n} \cdot \binom{2n-2}{n-1} x^{n-1}$$

$$= \sum_{n \geq 0} \frac{1}{n+1} \binom{2n}{n} x^n, \text{ as desired.}$$

Some problems involving permutations

For a permutation $p = (p_1, p_2, \dots, p_n)$, consider the graph formed by the edges $i \rightarrow p_i$. It is well-known that the graph is a union of several disjoint cycles.

Problem. Count the number of permutations of length n with k cycles.

These numbers are also called Stirling numbers of the first kind.

Let $c_n = (n-1)!$ be the number of permutations of length n which is a cycle. Let $C(x) = \sum_{n \geq 0} \frac{c_n}{n!} x^n$ denote the EGF of c . Let f_n be our answer and $F(x)$ be its EGF. The key observation here is that $F(x) = \frac{1}{k!} C(x)^k$.

Suppose for a moment our cycles are labelled from 1 to k . For every permutation, label each element with the label of the cycle it is in. Let's fix the length of cycle i to be a_i (so $\sum a_i = n$). Then, there are c_{a_i} ways to permute the elements in the i -th cycle and $\frac{n!}{a_1! a_2! \dots a_k!}$ ways to assign cycle labels to the elements of the permutation. Finally, in our actual problem, the order of cycles doesn't matter, so we need to divide by $k!$ in the end.

To summarize, the answer is $\frac{n!}{k!} \sum_{a_1+a_2+\dots+a_k=n} \frac{c_{a_1} c_{a_2} \dots c_{a_k}}{a_1! a_2! \dots a_k!}$. Verify that $\sum_{a_1+a_2+\dots+a_k=n} \frac{c_{a_1} c_{a_2} \dots c_{a_k}}{a_1! a_2! \dots a_k!}$ is $[x^n] C(x)^k$, so $F(x) = \frac{1}{k!} C(x)^k$ (the $n!$ disappears into $F(x)$ because we are dealing with EGFs).

Let's assume this is a CP problem and we are asked to find the answer for (n, k) . Then, we can calculate the answer directly using generating functions in $O(n \log n)$ since we can do exponentiation in $O(n \log n)$ ($P(x)^k = \exp(k \ln(P(x)))$).

Actually, what we just did is a special case of the more general Exponential Formula. However, I feel that it is easier to understand the Exponential Formula from these specific examples and you should try to understand it intuitively until it becomes common sense.

Problem. Count the number of permutations of length n such that all cycle lengths are in a fixed set of positive integers S .

We use the same trick as the previous problem, but let $c_i = 0$ if i is not in S .

This time, we need to find $[x^n] \sum_{k \geq 0} \frac{1}{k!} C(x)^k = [x^n] \exp(C(x))$ because we need to sum over all values of k (number of cycles), which can also be computed easily.

Problem. Find the expected number of cycles of a permutation of length n .

To compute the expected number of cycles, we count the sum of number of cycles over all permutations of length n . Let g_n denote the sum of number of cycles over all permutations of length n and $G(x)$ as the EGF of g . Using the same function C in the previous problems, we need to find (note the extra factor k which is the difference between this and the previous examples)

$$[x^n] G(x) = [x^n] \sum_{k \geq 0} \frac{k}{k!} C(x)^k = [x^n] C(x) \sum_{k \geq 1} \frac{1}{(k-1)!} C(x)^{k-1} = [x^n] C(x) \exp(C(x))$$

However, $C(x) = \sum_{k \geq 1} \frac{(k-1)!}{k!} x^k = \sum_{k \geq 1} \frac{x^k}{k} = -\ln(1-x)$. Hence, $C(x) \exp(C(x)) = -\frac{\ln(1-x)}{(1-x)}$.

For $n \geq 1$, $[x^n](-\ln(1-x)) = \frac{1}{n}$. By the Prefix Sum trick, $[x^n] \frac{-\ln(1-x)}{1-x} = 1 + \frac{1}{2} + \dots + \frac{1}{n}$.

Thus, $[x^n]G(x) = 1 + \frac{1}{2} + \dots + \frac{1}{n}$. Since $\frac{g_n}{n!}$ is the expected number of cycles of a permutation of length n , our answer is $1 + \frac{1}{2} + \dots + \frac{1}{n}$, the n -th Harmonic number!

We see that the exponential trick is viable when we are dealing with items that are made up from smaller pieces.

Stirling Numbers of the Second Kind

Problem. Find the number of ways to partition the set $\{1, 2, \dots, n\}$ into k subsets.

These numbers are also called Stirling numbers of the second kind.

Denote the answer by $f(n, k)$. The trick is to consider the polynomial (also known as **deck enumerator**) $D(x) = \sum_{n \geq 1} \frac{x^n}{n!}$. What is

$D(x)^k$? We have $[x^n]D(x)^k = \sum_{a_1+a_2+\dots+a_k=n, a_i \geq 1} \frac{1}{a_1!a_2!\dots a_k!}$. This sum has a similar combinatorial interpretation as the ones in the previous problems. Let's assume the partition sets are labelled from 1 to k . Then, a_i denotes the size of the i -th set and there are $\frac{n!}{a_1!a_2!\dots a_k!}$ ways to assign a set to each element (by the multinomial theorem). However, we have counted each partition $k!$ times, since in our final answer the sets shouldn't be ordered. Thus, $k!f(n, k) = n![x^n]D(x)^k$.

Rearranging gives us $\frac{f(n, k)}{n!} = \frac{[x^n]D(x)^k}{k!}$. Hence, $\sum_{n \geq 0} \frac{f(n, k)}{n!} x^n = \frac{D(x)^k}{k!}$. Introducing the variable y to correspond to the variable k , we have $\sum_{k \geq 0} \sum_{n \geq 0} \frac{f(n, k)}{n!} x^n y^k = \sum_{k \geq 0} \frac{[D(x)y]^k}{k!} = \exp(D(x)y)$.

Note: The polynomial $H(x, y) = \sum_{k \geq 0} \sum_{n \geq 0} f(n, k) \frac{x^n}{n!} y^k$ is also known as a **hand enumerator**.

Thus, we have the simple formula $H(x, y) = \exp(D(x)y)$. Note that $D(x) = \sum_{n \geq 1} \frac{x^n}{n!} = e^x - 1$, so we have $H(x, y) = e^{(e^x-1)y}$ (note how similar this is to the EGF of Bell numbers! In fact $H(x, 1)$ is the EGF of Bell numbers (can you see why?)).

To get the answer, we just need to find $n![x^n y^k]H(x, y) = n![x^n] \frac{(e^x-1)^k}{k!}$ which you can compute using polynomial operations efficiently.

Graph Counting

Problem. Find the number of vertex-labeled undirected graphs with n vertices so that each vertex has degree 2.

Every such graph must be a union of disjoint cycles (why?). As usual, we start by considering the generating function for one "component" in the item we need to count. Let d_n denote the number of **undirected** cycles of length n and $D(x)$ denote its EGF. Then,

$$D(x) = \sum_{n \geq 3} \frac{(n-1)!}{2n!} x^n = \frac{1}{2} \sum_{n \geq 3} \frac{x^n}{n} = \frac{1}{2} \left(-\ln(1-x) - x - \frac{x^2}{2} \right).$$

Let $G(x)$ denote the EGF of our answer. Using the same argument as before, we find that $G(x) = \exp(D(x))$, so we get the formula

$$G(x) = \exp \left(\ln \left(\sqrt{\frac{1}{1-x}} \right) - \frac{x}{2} - \frac{x^2}{4} \right) = \frac{e^{-\frac{x}{2} - \frac{x^2}{4}}}{\sqrt{1-x}}, \text{ and you can compute the coefficient of } x^n \text{ to obtain the answer.}$$

Let's look at a trickier example.

Problem. Find the number of bipartite vertex-labeled graphs with n vertices.

It is tempting to try a similar approach as the previous problem. We can relate the number of bipartite graphs with the number of connected bipartite graphs. Can we count the number of connected bipartite graphs easily? Unfortunately, it does not seem to be too easy to count.

Instead, let us color each vertex of the graph with red or blue, and count the number of **colored** bipartite graphs (not necessarily connected). Suppose we choose k vertices to color red and $n - k$ to color blue. Then, there are $\binom{n}{k}$ ways to choose the coloring and $2^{k(n-k)}$ to choose the edges (since each edge must be between 2 components). Thus, the number of colored bipartite graphs on n vertices is

$\sum_{k \geq 0} \binom{n}{k} 2^{k(n-k)}$. Call this number a_n and its EGF as $A(x)$.

The next step is to relate the number of colored bipartite graphs with the number of colored connected bipartite graphs. Let b_n denote the number of colored connected bipartite graphs on n vertices and $B(x)$ be its EGF. Using a similar argument as before, we have the relation $A(x) = \exp(B(x))$, and thus $B(x) = \ln(A(x))$.

Returning to our original problem, our next step is to count the number of connected bipartite graphs on n vertices (call the count c_n and EGF $C(x)$). However this is easy, since each connected bipartite graph can be colored in exactly two ways (the coloring is fixed once we choose the color of a vertex). Hence, $C(x) = \frac{B(x)}{2}$.

Finally, let d_n be the number of bipartite graphs on n vertices and $D(x)$ be its EGF. Then, we have $D(x) = \exp(C(x))$ using the exponential argument. Thus, $D(x) = \exp(C(x)) = \exp\left(\frac{B(x)}{2}\right) = \exp\left(\frac{\ln(A(x))}{2}\right) = \sqrt{A(x)}$, which is a nice formula!

Placing Rooks and PIE

Let's look at a last example which demonstrates the use of the Inclusion-Exclusion Principle (PIE).

Consider a $n \times n$ chessboard where some cells are colored black and others are colored white. Suppose we magically know the sequence r_k , the number of ways to place k non-attacking rooks on white cells of the chessboard (i.e. no two are on the same row or column, no rooks are on black cells). Let e_k denote the number of ways to place n non-attacking rooks on the chessboard so that exactly k of the rooks are on white squares. Can we find e_k in terms of r_k ?

The trick is that exact conditions are usually harder to count while "at least" conditions are easier to count. For a fixed subset of white cells S , denote $N(S)$ as the number of ways to place n non-attacking rooks on the chessboard such that there is at least one rook on each cell in S .

Let $n_k = \sum_{|S|=k} N(S)$.

We relate n_k with e_k . Consider a subset T of size t and a way to place n non-attacking rooks so that the white cells they occupy is exactly T . Every k -element subset of T contributes to the sum n_k . Thus, we obtain the recurrence $n_k = \sum_{t \geq 0} \binom{t}{k} e_t$.

Let $N(x)$ and $E(x)$ be the OGFs of n_k and e_k . We can derive a simple relation between $N(x)$ and $E(x)$. Indeed, we have

$$N(x) = \sum_{k \geq 0} x^k \sum_{t \geq 0} \binom{t}{k} e_t = \sum_{t \geq 0} e_t \sum_{k \geq 0} \binom{t}{k} x^k = \sum_{t \geq 0} e_t (x+1)^t = E(x+1).$$

Thus, we have the simple relation $E(x) = N(x-1)$.

It turns out that n_k is usually much easier to find. In our problem, $n_k = r_k(n-k)!$, since we can first choose our set S as any set of k non-attacking rooks on white cells, then place the other $n-k$ rooks in $(n-k)!$ ways. Thus, we obtain $N(x) = \sum_{k \geq 0} r_k(n-k)!x^k$ and

$E(x) = \sum_{k \geq 0} r_k(n-k)!(x-1)^k$. Hence, we can read e_j from the coefficients of $E(x)$.

Proving Some Interesting Theorems via Generating Functions

This is not entirely CP related but here are some cool theorems you can prove easily with generating functions.

Partition in odd parts = Partition in distinct parts

A partition of n into k parts is a multiset of positive integers of size k which sum up to n . For example, $\{3, 1, 1\}$ is a partition of 5 into 3 parts. Note that the order of elements do not matter, so $\{3, 1, 1\}$ and $\{1, 3, 1\}$ are the same partition.

You might have heard of the well-known problem of proving that the number of partitions of n into odd parts is equal to the number of partitions of n into distinct parts. Here, we prove a generalization of it.

Problem. Prove that the number of partitions of n into parts of size not divisible by $k + 1$ is equal to the number of partitions of n into parts such that there are at most k parts of each size.

Note that when $k = 1$ we reduce this to the standard problem.

Fix k and let $A(x)$ be the OGF of the first object and $B(x)$ be the OGF of the second object. Observe that choosing a partition is the same as choosing the number of times we use each integer in our multiset, so

$$\begin{aligned} B(x) &= \prod_{r \geq 1} (1 + x^r + x^{2r} + \dots + x^{kr}) \\ &= \prod_{r \geq 1} \left(\frac{1 - x^{r(k+1)}}{1 - x^r} \right) \\ &= \prod_{r \geq 1, (k+1) \nmid r} \left(\frac{1}{1 - x^r} \right) \\ &= \prod_{r \geq 1, (k+1) \nmid r} (1 + x^r + x^{2r} + \dots) = A(x) \end{aligned}$$

Binet's Formula (and solving Linear Recurrences)

Let f_n denote the n -th Fibonacci number (with $f_0 = 0$, $f_1 = 1$, $f_n = f_{n-1} + f_{n-2}$). You may have heard of Binet's Formula, which states that $f_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right]$.

This might look very random, but it actually comes directly from the generating function. Recall that $F(x) = \frac{x}{1-x-x^2}$ is the OGF of f . The trick here is to use **partial fractions** (we will explore another example in the next part). Let $-\gamma_1, -\gamma_2$ be the roots of the equation $1 - x - x^2 = 0$ (where $\gamma_1 = \frac{1+\sqrt{5}}{2}$, $\gamma_2 = \frac{1-\sqrt{5}}{2}$). Then, we can write $F(x) = \frac{A}{x+\gamma_1} + \frac{B}{x+\gamma_2}$. With some calculation, we can obtain

$$F(x) = \frac{1}{\gamma_1 - \gamma_2} \left(\frac{1}{1 - \gamma_1 x} - \frac{1}{1 - \gamma_2 x} \right) = \frac{1}{\sqrt{5}} \left(\sum_{j \geq 0} \gamma_1^j x^j - \sum_{j \geq 0} \gamma_2^j x^j \right).$$

Comparing coefficients, we get $f_n = \frac{1}{\sqrt{5}} (\gamma_1^n - \gamma_2^n)$. Recalling that $\gamma_1 = \frac{1+\sqrt{5}}{2}$ and $\gamma_2 = \frac{1-\sqrt{5}}{2}$ gives us Binet's Formula.

Note that this method is generalizable for general linear recurrences.

Probability that a random permutation has no cycle length which is the square of an integer

Problem. What is the probability that a random permutation has no cycle length which is the square of an integer?

This problem seems pretty random (no pun intended), but I include it here to show the power of generating functions.

Firstly, suppose we know that our permutation is of length n and there are a_i cycles of length i (so $\sum_{i \geq 1} i a_i = n$). How many such

permutations are there? With some simple counting, we can obtain the formula
$$\frac{n! \prod_{i \geq 1} ((i-1)!)^{a_i}}{\prod_{i \geq 1} (i!)^{a_i} \cdot \prod_{i \geq 1} (a_i!)} = \frac{n!}{\prod_{i \geq 1} i^{a_i} \cdot \prod_{i \geq 1} (a_i!)} \quad (\text{Hint: Assume$$

the cycles are labelled first, and assign the elements into cycles, then arrange the elements within each cycle. Divide some factorials to handle overcounts).

The sequence $a = (a_1, a_2, \dots)$ defined above is also called the **cycle type** of a permutation.

Let $c(a)$ denote the number of permutations of length $n = a_1 + 2a_2 + \dots$ with cycle type a and $p(a)$ denote the probability that a permutation of length $n = a_1 + 2a_2 + \dots$ has cycle type a . Hence, $c(a) = \frac{n!}{\prod_{i \geq 1} i^{a_i} \cdot \prod_{i \geq 1} (a_i!)}$ and $p(a) = \frac{c(a)}{n!}$

Now, the trick is to consider the **infinite-variable** generating function in x_1, x_2, \dots :

$$C(x, y) = \sum_{n \geq 0} \frac{y^n}{n!} \sum_{a_1 + 2a_2 + \dots = n, a_i \geq 0} c(a) x_1^{a_1} x_2^{a_2} \dots$$

From our discussion above, we know how to find $c(a)$, thus we can write $C(x, y)$ as

$$C(x, y) = \left(\sum_{a_1 \geq 0} \frac{(yx_1)^{a_1}}{a_1! 1^{a_1}} \right) \left(\sum_{a_2 \geq 0} \frac{(y^2 x_2)^{a_2}}{a_2! 2^{a_2}} \right) \dots = \exp(yx_1) \exp(y^2 \frac{x_2}{2}) \dots = \exp\left(\sum_{i \geq 1} \frac{y^i x_i}{i}\right).$$

Hence, for a fixed cycle type $a = (a_1, a_2, \dots)$, $p(a) = [x_1^{a_1} x_2^{a_2} \dots] \exp\left(\sum_{i \geq 1} \frac{y^i x_i}{i}\right)$.

Let us return to our problem. Call a cycle type a good if $a_j = 0$ for all perfect squares j . We want to find

$\lim_{n \rightarrow \infty} \sum_{a \text{ good}} [y^n x_1^{a_1} x_2^{a_2} \dots] \exp\left(\sum_{i \geq 1} \frac{y^i x_i}{i}\right)$. We can "substitute" $x_j = 1$ for all non-perfect squares j to indicate that we don't care about the power of x_j if j is not a perfect square. so we reduce our problem to finding (noting that $a_j = 0$ for all perfect squares j)

$$\begin{aligned} & \lim_{n \rightarrow \infty} [y^n] \exp\left(\sum_{i=z^2} \frac{y^i x_i}{i} + \sum_{i \neq z^2} \frac{y^i}{i}\right) \\ &= \lim_{n \rightarrow \infty} [y^n] \exp\left(\sum_{i=z^2} \frac{y^i (x_i - 1)}{i} + \sum_{i \geq 1} \frac{y^i}{i}\right) \\ &= \lim_{n \rightarrow \infty} [y^n] \exp\left(\sum_{i=z^2} \frac{y^i (x_i - 1)}{i} - \ln(1 - y)\right) \\ &= \lim_{n \rightarrow \infty} [y^n] \frac{1}{1 - y} \exp\left(\sum_{i=z^2} \frac{y^i (x_i - 1)}{i}\right) \end{aligned}$$

If we let $A(y)$ be the OGF of $a_n = [y^n] \exp\left(\sum_{i=z^2} \frac{y^i (x_i - 1)}{i}\right)$, then by the Prefix Sum trick, our limit is equal to $\lim_{n \rightarrow \infty} \sum_{i \geq 0} a_i$ (assuming the sum converges).

Intuitively, we can get the "sum to infinity" by substituting $y = 1$ into $\exp\left(\sum_{i=z^2} \frac{y^i (x_i - 1)}{i}\right)$, and we are only interested in the terms without x_i (for i a perfect square), so we let these $x_i = 0$, to obtain

$$\lim_{n \rightarrow \infty} [y^n] \frac{1}{1 - y} \exp\left(\sum_{i=z^2} \frac{y^i (x_i - 1)}{i}\right) = \exp\left(\sum_{i=z^2} -\frac{1}{i}\right) = e^{-\frac{\pi^2}{6}}, \text{ which is actually our answer (recall that } \sum_{i \geq 1} \frac{1}{i^2} = \frac{\pi^2}{6} \text{).}$$

Snake Oil Trick in Proving (or Finding) Combinatorial Identities

To end the first part of this tutorial, I will briefly introduce a trick to simplify combinatorial identities using generating functions. The idea is that instead of dealing with the sum directly, it is easier to deal with the series obtained from the generating functions.

Problem. Find the sum $\sum_{k \geq 0} \binom{k}{n-k}$ for a fixed positive integer n .

Suppose the answer to our problem is $f(n)$. The idea is that it is easier to consider the OGF of f , which is $F(x) = \sum_{n \geq 0} f(n)x^n$. We have

$$\begin{aligned} F(x) &= \sum_{n \geq 0} f(n)x^n \\ &= \sum_{n \geq 0} \sum_{k \geq 0} \binom{k}{n-k} x^n \end{aligned}$$

Now, we switch summation signs to obtain

$$= \sum_{k \geq 0} \sum_{n \geq 0} \binom{k}{n-k} x^n$$

The key idea is to make the inner sum easy to compute, and we know how to compute $\sum_{n \geq 0} \binom{k}{n-k} x^{n-k}$, since it is just $\sum_{r \geq 0} \binom{k}{r} x^r$ in disguise with $r = n - k$!

Thus, we factor out x^k to obtain

$$\begin{aligned} &= \sum_{k \geq 0} x^k \sum_{n \geq 0} \binom{k}{n-k} x^{n-k} \\ &= \sum_{k \geq 0} x^k (1+x)^k \\ &= \sum_{k \geq 0} (x(1+x))^k \\ &= \frac{1}{1-x-x^2} \end{aligned}$$

Do you recognize the last expression? It is actually $\frac{1}{x} F(x)$ where $F(x)$ is the OGF of the Fibonacci numbers! Thus, $\frac{1}{1-x-x^2} = \sum_{n \geq 0} f_{n+1} x^n$, and by comparing coefficients we get $f(n) = f_{n+1}$, the $(n+1)$ -th Fibonacci number!

There are many other similar applications of the Snake Oil Method but I won't go into detail here. In general, this method might be useful in CP if you encounter some math problems and reduce the problem into double or triple summation of binomial coefficients but you need an $O(n)$ solution. Sometimes, you can forcefully simplify your summations using the Snake Oil method. We will use the trick of introducing a power series and swapping summation signs again to simplify some expressions in the next part of this article.

As an exercise, try to prove the following identity with the Snake Oil method:

$\sum_{r=0}^k \binom{m}{r} \binom{n}{k-r} = \binom{n+m}{k}$ (there is an obvious bijective proof, can you see why this is true?). This identity is very useful in simplifying sums involving binomial coefficients.

This ends the first part of my tutorial on generating functions. The next part will focus more on applications on GFs in CP problems, so stay tuned!

UPD: Part 2 is now available [here](#).

P.S. Let me know if I made any errors or typos in the post (which is likely to happen).

mathforces, #basic math, generating function, tutorial