浅谈几种基础的生成函数

江西师范大学附属中学信息学竞赛组
2021 级（17）班黄晨曦

2019 年 10 月 7 日

1 简述

计数问题是信息学竞赛中的热点问题之一，其解法也非常多样。而生成函数能解决的问题非常的多，很多题目都来自于生成函数思想的启发。本文将用几个简单的例子来介绍两种简单的生成函数：普通生成函数（Ordinary Generating Function, OGF）和指数型生成函数（Exponential Generating Function, EGF）。由于本人的学识浅薄，且为业余的离散数学爱好者，所以对于文中之纰漏还请读者海涵。由于信息学竞赛的特性，有一些非常复杂的证明我并不会列入文中，有兴趣的读者可以在其他时间进行相关资料的查阅。

2 预备知识

2.1 一些扩展定理

定义 1「扩展组合数」(或「广义组合数」) 把组合数的定义推广到了实数域上了:

$$\binom{n}{k} = \frac{n(n-1)(n-2)\dots(n-k+1}{k!}, n \in \mathbb{R}, k \in \mathbb{N}$$

定理 1「扩展二项式定理」(或「广义二项式定理」) 把狭义组合数换成了广义组合数，最后结果仍然满足:

$$(1+z)^{\alpha} = \sum_{k=0}^{\infty} \binom{\alpha}{k} z^k$$

2.2 几个麦克劳林级数

2.2.1 几何级数

几何级数指的是形如无限项等比数列的级数:

$$g(x) = \sum_{i=0}^{\infty} x^i = \frac{1}{1-x}$$

证明 2.1 根据扩展二项式定理，有:

$$(1+z)^{\alpha} = \sum_{k=0}^{\infty} \binom{\alpha}{k} z^k$$

令 $\alpha = -n$，带入原式:

$$(1+z)^{-n} = \frac{1}{(1+z)^n} = \sum_{k=0}^{\infty} \binom{-n}{k} z^k$$

根据扩展组合数的定义，可以化简这个式子:

$$\binom{n}{k} = \frac{n(n-1)(n-2)\dots(n-k+1)}{k!}, n \in \mathbb{R}, k \in \mathbb{N}$$

$$\begin{aligned}\sum_{k=0}^{\infty}\binom{-n}{k}z^k &= \sum_{k=0}^{\infty}\frac{-n(-n-1)(-n-2)\dots(-n-k+1)}{k!}z^k \\ &= \sum_{k=0}^{\infty}(-1)^k\frac{n(n+1)(n+2)\dots(n+k-1)}{k!}z^k \\ &= \sum_{k=0}^{\infty}(-1)^k\binom{n+k-1}{k}z^k\end{aligned}$$

然后，我们可以尝试去让 $z=-x$，带入式子：

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

令 $n=1$，然后就结束了。

$$\frac{1}{1-x} = \sum_{k=0}^{\infty}\binom{k}{k}x^k = \sum_{k=0}^{\infty}x^k$$

证毕。

几何级数的推导内容在一般生成函数中非常常见。以下级数的推导都来自于上面的证明过程，它们的推导非常的显然：

$$\begin{aligned}g(x) &= \sum_{k=0}^{\infty}a^kx^k = \sum_{k=0}^{\infty}(ax)^k = \frac{1}{1-ax} \\ h(x) &= \sum_{k=0}^{\infty}\binom{n}{k}x^k = (1+x)^n \\ f(x) &= \sum_{k=0}^{\infty}(-1)^kx^k = \frac{1}{1+x}\end{aligned}$$

2.2.2 指数级数

指数级数指的就是形如 $g^{(e)}(x)$ 的级数：

$$g^{(e)}(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!} = e^x$$

指数级数所对应的就是 e^x 的泰勒展开，具体证明读者可以对指数函数的泰勒展开进行了解。

3 普通生成函数

普通生成函数是一类常用的生成函数。对于一个数列 $\{a_i\}$，其生成函数是：

$$f(x) = \sum_{n=0}^{\infty} a_n x^n$$

在预备知识中，我们提到了几何级数。所以，从生成函数的角度去研究，发现几何级数对应的其实就是常数列：

$$\{a_n\}, \forall a_n = 1 \to g(x) = \sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$$

接下来我们来看几个经典的计数序列的生成函数。

3.1 不定方程非负解的计数问题

我们设 h_n 为以下这个不定方程的解的方案数：

$$\sum_{i=1}^{k} a_i = n$$

显然可以解出：

$$h_n = \binom{n+k-1}{k-1}$$

我们可以自然地写出它的生成函数：

$$g(x) = \sum_{i=0}^{\infty} \binom{i+k-1}{k-1} x^i$$

根据预备知识中的对几何级数的推导可找到最简形式。简单回顾一下：

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

发现这个函数的形式其实就是几何级数的 k 次方。仔细思考生成函数乘法的意义，对于该生成函数的第 n 项，其实就是两个多项式中恰好能组成 n 的系数之和。提出一项 a_nx^n 进行分析：

$$a_nx^n = \sum_{i=0}^{n}(A_ix^i)(B_{n-i}x^{n-i}) = \sum_{i=0}^{n}A_iB_{n-i}x^n$$

所以，多项式的乘法后，每一项都可以写作前两个多项式中某些项的卷积。在信息学竞赛中，这可以和快速傅立叶变换（FFT）相结合，在许多题目中运用广泛。

回到数列 $\{h_n\}$ 的生成函数：发现做了 k 次几何级数的多项式乘法，其实就是把可能解的方案数全部记录了，因为不定方程非负解的个数显然符合多个多项式卷积的形式。这就是生成函数的计数原理。

3.2 背包型计数问题的生成函数

背包型计数问题是生成函数的经典案例。如果我们对于 k 个类别的物品，价格为 a_i，每一种都有无限多，我们如何来计算总价格 n 的方案数呢?

我们考虑让次数为 n 的那一个单项式的系数为方案数，设为 $\{b_n\}$。那么，我们为了拼成 n，对于每一个物品都可以设置一个生成函数 $f_i(x)$：

$$g_i(x) = 1 + x^{b_i} + x^{2b_i} + x^{3b_i} + \cdots = \sum_{p=0}^{\infty}(x^{b_i})^p = \frac{1}{1-x^{b_i}}$$

对于最终的计数序列的生成函数 $f(x)$，其实就是这些物品的生成函数之积。请读者仔细思考背包型生成函数的计数原理，加深对于生成函数计数的原理的印象。

$$f(x) = \prod_{i=1}^{k}g_i(x)$$

那么对于物品的个数为 c_i 的情况下，那我们直接把单个生成函数的项数改成有穷的即可：

$$g_i(x) = \sum_{p=0}^{c_i} (x^{b_i})^p = \frac{1}{1-x^{b_i}} \pmod{x^{b_i c_i+1}}$$

3.3 生成函数与卷积

回顾之前不定方程非负解的生成函数函数和背包型计数问题的生成函数，我们发现其计数原理都利用到了多项式乘法的原理，其实也就是卷积：

$$a_n x^n = \sum_{i=0}^{n} (A_i x^i)(B_{n-i} x^{n-i}) = (\sum_{i=0}^{n} A_i B_{n-i}) x^n$$

很多计数问题的计数式子都含有卷积的形式，这个时候就可以使用多项式乘法来快速求的前 n 项的值。接下来是一道例题，来自「集训队互测 2013| 城市规划」：

你需要求出 n 个点的简单（无重边无自环）无向连通图数目。答案对 1004535809 取模。

解这题的关键在于推出正确的递推公式，设置 f_n 为答案，g_n 为无向图的个数。根据定义可以推导出：

$$\sum_{i=1}^{n} f_i \binom{n-1}{i-1} g_{n-i} = g_n$$

g_n 显然为 $2^{\binom{n}{2}}$。那么整理式子可以得到：

$$\sum_{i=1}^{n} \frac{f_i}{(i-1)!} \frac{g_{n-i}}{(n-i)!} = \frac{g_n}{(n-1)!}$$

发现等式左边为卷积的形式。所以我们可以直接把这个计数序列的求解问题转移到多项式上进行：

$$G(x) = \sum_{i=0}^{\infty} \frac{g_i}{i!} x^n$$
$$C(x) = \sum_{i=0}^{\infty} \frac{g_i}{(i-1)!} x^n$$
$$F(x) = \sum_{i=0}^{\infty} \frac{f_i}{(i-1)!} x^n$$

最后的生成函数就是：

$$F(x)G(x) \equiv C(x) \pmod{x^{n+1}}$$
$$F(x) \equiv C(x)G^{-1}(x) \pmod{x^{n+1}}$$

对这个生成函数进行时间复杂度为 $\Theta(n \log n)$ 的快速傅立叶变换即可解出整个序列。