



铃悬 的博客

铃悬的数学小讲堂——生成函数进阶与简单的图计数

2019-05-01 19:06:43

👍 29

PS: 看不懂的可以先跳过去...不过好像这篇里面没有相对主线来说太难的东西。

这篇之前, 你大概需要先了解生成函数基础内容, 比如“[生成函数初步](#)”那一篇。

-1. 符号及约定

1. $[P]$ (艾弗森括号) 是指, 若 P 为真, 则 $[P] = 1$, 否则 $[P] = 0$ 。即 $[P] = \begin{cases} 1 & P \\ 0 & \neg P \end{cases}$ 。特别的, 若 P 为假, $[P] \times e$ 即使在 e 未定义时也为 0 (例如, $[n \neq 0] \frac{1}{n}$ 在 $n = 0$ 是为 0, 而不是未定义值)。
2. 大写字母代表一个函数, 例如 $F(x)$ 代表 $\{f_n\}$ 的生成函数。有时为方便整洁会省略参数列表, 即 $F(x)$ 省略为 F 。
3. 若无特殊约定, 数列下标均从 0 开始。
4. 斜体表示吐槽, 你完全不需要看明白。
5. $x^0 = 1$ 对所有数成立, 包括 0。

0. 前备知识

3. 微积分初步。大概知道怎么求导数（包括初等函数、四则运算和复合）、以及简单的积分就可以了...

1. 指数型生成函数

1.1 定义

在处理一些计数问题的时候，由于物品是有标号的，在计数会求这么一个类似卷积的东西：

$$h_n = \sum_{i=0}^n \binom{n}{i} f_i g_{n-i}$$

暂且把它叫做二项卷积（并没有找到这东西的名字qaq）。可以看到它和普通卷积的区别在于乘上了一个组合数。举个例子，如果要把 n 个球放到两个盒子里，而第一个盒子若放 i 个球就会有 f_i 种方式涂色，第二个盒子若放 i 个球就会有 g_i 种涂色方式（不要问我为什么盒子涂色还跟里面的球有关），那么把 n 个球放入两个盒子并给两个盒子涂色的方案数就是上面的 h_n 。

把组合数展开，提出公共项 $n!$ 并除到左边，就会得到

$$\frac{h_n}{n!} = \sum_{i=0}^n \frac{f_i}{i!} \cdot \frac{g_{n-i}}{(n-i)!}$$

这样的话，相当于说 $\{h_n/n!\}$ 是 $\{f_n/n!\}$ 和 $\{g_n/n!\}$ 的卷积。那么可以对 $\{f_n/n!\}$ 搞出生成函数，即

$$F(x) = \sum_{n \geq 0} f_n \frac{x^n}{n!}$$

称这个生成函数为数列 $\{f_n\}$ 的 **指数型生成函数**（Exponential Generating Function, EGF）。可以发现二项式卷积得到的 $\{h_n\}$ 的 EGF 就是将 $\{f_n\}$ 和 $\{g_n\}$ 的 EGF 乘起来。

如果没有特殊点出的话，本篇文章以下的函数均为 EGF（而不是 OGF）。

$$\begin{aligned} \left(\sum_{n \geq 0} f_n \frac{x^n}{n!} \right) \pm \left(\sum_{n \geq 0} g_n \frac{x^n}{n!} \right) &= \sum_{n \geq 0} (f_n \pm g_n) \frac{x^n}{n!} \\ c \times \left(\sum_{n \geq 0} f_n \frac{x^n}{n!} \right) &= \sum_{n \geq 0} c f_n \frac{x^n}{n!} \\ \left(\sum_{n \geq 0} f_n \frac{x^n}{n!} \right) \left(\sum_{n \geq 0} g_n \frac{x^n}{n!} \right) &= \sum_{n \geq 0} \left(\sum_{i=0}^n \binom{n}{i} f_i g_{n-i} \right) \frac{x^n}{n!} \end{aligned}$$

指数型生成函数的加减法显然还是逐项加减，而乘法就对应二项卷积（见上）。这些都是可以直接验证的。

如果把一个 EGF 求导或积分，就会得到

$$\begin{aligned} \frac{d}{dx} \sum_{n \geq 0} f_n \frac{x^n}{n!} &= \sum_{n \geq 0} f_{n+1} \frac{x^n}{n!} \\ \int \sum_{n \geq 0} f_n \frac{x^n}{n!} dx &= \sum_{n \geq 1} f_{n-1} \frac{x^n}{n!} + C \end{aligned}$$

因为 $\frac{x^n}{n!}$ 求导得到 $\frac{x^{n-1}}{(n-1)!}$ ，而积分得到 $\frac{x^{n+1}}{(n+1)!}$ 。

如果像 OGF 一样乘或除以 x ，就会得到类似 OGF 的求导或积分一样的东西：

$$\begin{aligned} x \sum_{n \geq 0} f_n \frac{x^n}{n!} &= \sum_{n \geq 1} \frac{f_{n-1}}{n} \cdot \frac{x^n}{n!} \\ \frac{1}{x} \left(\sum_{n \geq 0} f_n \frac{x^n}{n!} - f_0 \right) &= \sum_{n \geq 0} (n+1) f_{n+1} \frac{x^n}{n!} \end{aligned}$$

1.3 简单的例子

$$\sum_{n \geq 0} c^n \frac{x^n}{n!} = e^{cx}$$

$$\sum_{n \geq 0} n^k \frac{x^n}{n!} = x^k e^x$$

$$\sum_{n > 0} (n-1)! \frac{x^n}{n!} = \ln \frac{1}{1-x}$$

当然这些都可以直接用 $\{f_n/n!\}$ 的 OGF 来得到...

2. 贝尔数——exp

定义贝尔数 w_n , 表示把 n 个有区别的小球放到若干个 (任意多个) 无区别的盒子里的方案数。换句话说就是把集合 $\{1, 2, \dots, n\}$ 划分成若干个不相交非空子集的并的方案数。比如 $w_3 = 5$, 因为 $\{1, 2, 3\} = \{1\} \cup \{2\} \cup \{3\} = \{1\} \cup \{2, 3\} = \{1, 2\} \cup \{3\} = \{1, 3\} \cup \{2\}$ (注意都放到一个盒子里, 即只划分成一个子集, 也是允许的)。

对于贝尔数的递推, 可以考虑枚举 1 所在的子集大小。如果 1 所在的子集包含 i 个元素, 那么还要从剩下 $n-1$ 个里面选出 $i-1$ 个和 1 放在同一个集合, 而其他 $n-i$ 个有 w_{n-i} 种放法。因此

$$w_n = [n=0] + \sum_{i=1}^n \binom{n-1}{i-1} w_{n-i}$$

可以发现等号右侧就是数列 $\{w_n\}$ 和 $\{1\}$ 的二项卷积的第 $n-1$ 项。因此可以将两者的 EGF ($W(x)$ 和 e^x) 乘起来再向右移一位, 即积分一下。所以可以得到

$$W(x) = 1 + \int W(x) e^x dx$$

两边求导,

$$\frac{dW(x)}{dx} = W(x) e^x$$

$$\frac{dW(x)}{W(x)} = e^x dx$$

$$\ln W(x) = e^x + C$$

$$W(x) = e^{e^x - 1} = \exp(e^x - 1)$$

这个式子看上去比之前的递推公式简洁很多，它应该有一个更优美的组合解释。

考虑到 $A(x) = e^x - 1$ 就是“非空子集”的 EGF（因为对于任何的 $n > 0$ ，都恰有一个方案把 n 个元素放到一个集合里（废话）），那么在外面套一层 \exp ？展开 \exp 的式子

$$\exp A(x) = \sum_{k \geq 0} \frac{A^k(x)}{k!}$$

将这里枚举的 k 看做非空子集的个数，那么对应的把元素放到 k 个非空子集里的方案数对应的 EGF 就是 $A^k(x)$ 。至于除以 $k!$ ，则是因为这 k 个子集是无序的。

对于其他东西而不只是非空子集，也可以由类似的解释搞出一个 \exp 。抽象一下，就是要把 n 个有标号的球放到任意多个无标号的盒子里，并且一个放有 k 个球的盒子有 a_k 种染色方案，那么放球并染色的方案就是 $[x^n] \exp A(x)$ 。

另外，那个递推式可以看成 $\frac{d(\exp A(x))}{dx} = \frac{dA(x)}{dx} \exp A(x)$ 。

至于这东西怎么求...好吧这篇文章只是讲理论的，可以参考多项式算法模板之类的...

3. 简单无向连通图——ln

求 f_n ，表示 n 个点的简单无向连通图的个数（当然，点是带标号的）。

先令 g_n 表示 n 个点的简单无向图的个数，即 $2^{n(n-1)/2}$ 。

要求 f_n ，可以用 g_n 减去那些不连通图的个数。不连通的话，枚举 1 号点所在的连通块大小 i ，需要先 $\binom{n-1}{i-1}$ 选出和 1 在一个连通块的方案数，再乘上 f_i 把 i 个点连成一个连通块， g_{n-i} 剩下的点任意连。

那么

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

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

换个角度思考一下，如果我们知道 f_n 要求 g_n ，就是像上面贝尔数一样求。

这样可以得到 $G(x) = \exp F(x)$ ，或 $F(x) = \ln G(x)$ 。

于是我们得到了 \ln 的用处：如果我们知道了一个生成函数 \exp 之后是什么，就可以把它 \ln 回来...

4. P4233 强连通竞赛图计数——求逆

“竞赛图”是指任意两个点之间都恰好有一条边的有向图。求 n 个点的强连通竞赛图的个数。

P4233 原题是求所有有哈密顿回路的竞赛图的哈密顿回路个数的期望。但是实际上所有的哈密顿回路总数很好求（就是 $(n-1)!2^{n(n-1)/2-n}$ 条），所以实际上就是要求具有哈密顿回路的竞赛图的个数，而后者又可以等价于强连通竞赛图（至于为什么强连通竞赛图必有哈密顿回路，实际上挺好证的...您们自己想吧）

我们发现强连通竞赛图不好求，但是任意竞赛图很好求。那么根据上一题的想法，可以先用强连通竞赛图的个数表示出任意竞赛图的个数，然后反推回去。

令 f_n 表示 n 个点的强连通竞赛图个数， g_n 表示竞赛图的个数。

要用强连通竞赛图表示任意竞赛图，可以考虑枚举这个竞赛图的拓扑序最小的强连通分量的大小 i （因为可以发现竞赛图缩强连通分量之后一定是一条链）。由于这个强连通分量和其他点之间的边的方向是一定的（必定是从这 i 个点连向其他 $n-i$ 个点），所以方案数就是 $f_i g_{n-i}$ 。于是

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

这看上去好像就是说 g 是它自己与 f 的二项卷积。于是搞出两者的 EGF $G(x), F(x)$ ，就有

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

于是

$$F(x) = 1 - \frac{1}{G(x)}$$

至于多项式的（乘法）逆怎么算...还是去看多项式板子吧...

5. P4389 无穷背包方案数——ln/exp

P4389

有一个无穷背包，也就是说我们有若干个物品，每种物品都有无限个。

有 n 种物品，第 i 种的体积为 v_i 。求用物品恰好填满大小为 m 的背包的方案数。

考虑用 $f_{i,j}$ 表示前 i 种物品恰好填满 j 的体积的方案数，则

$$f_{i,j} = f_{i-1,j} + f_{i,j-v_i}$$

如果令 $F_i(x) = \sum_{j \geq 0} f_{i,j} x^j$ ，也就是 f_i 的 OGF（注意，这次 x 的次数是“体积”，这东西是没有标号的，因此我们使用 OGF 而非 EGF），那么

$$F_i(x) = \sum_{j \geq 0} (f_{i-1,j} + f_{i,j-v_i}) x^j = F_{i-1}(x) + x^{v_i} F_i(x)$$

也就是说

$$F_i(x) = \frac{1}{1 - x^{v_i}} F_{i-1}(x)$$

这样的话，可以发现最后的 $F(x) = F_n(x)$ 就是

$$\prod_{i=1}^n \frac{1}{1 - x^{v_i}}$$

前一篇文章的划分数对应的就是 $v_i = i, n \rightarrow \infty$ 的情况。这个式子也可以理解成直接把每种物品的 OGF 乘起来，因为两种背包合并就是一个通常的卷积。

那么如何计算这个东西呢？

$$\begin{aligned}
 &= \exp \ln \prod_{i=1}^n \frac{1}{1-x^{v_i}} \\
 &= \exp \sum_{i=1}^n \ln \frac{1}{1-x^{v_i}} \\
 &= \exp \sum_{i=1}^n \sum_{j \geq 1} \frac{x^{jv_i}}{j}
 \end{aligned}$$

如果我们把 v_i 相同的合并同类项，假设 $v_i = k$ 的有 b_k 个，那这个式子就变成了

$$\exp \sum_{k \geq 1} b_k \sum_{j \geq 1} \frac{x^{jk}}{j}$$

因为 \exp 的前 m 项只依赖于原幂级数的前 m 项，所以只需要算那些 $jk \leq m$ 的项就可以了。众所周知这样直接枚举 j, k 的复杂度是 $O(m \log m)$ 的。这样本题就完美解决了。

这样的话，就可以发现 \ln / \exp 的另一个用途，即在 \prod 和 \sum 之间进行转换。

6. 总结

又是很快就结束了。由于目前 OI 中对于生成函数的应用并不多，笔者也没有找到更多的难度适宜的例题（并且这里主要是想要讲一点简单的图计数，虽然确实很简单）；而且这篇博客（以及前一篇，即生成函数初步）更多的是想要让没有接触过的同学了解一些基础的内容而不至于面对一些题解而感到摸不到头脑，所以笔者认为这些内容已经足够。

敬请期待本篇的续集——**困难的图计数**



我草居然是 rgy，僭越了。。但确实应该是写错了