

生成函数

这几周比较忙，还有很多写得不详细的地方。
本文的知识点和例题可能比较片面。

形式幂级数

$A(x) = \sum_{i \geq 0} a_i x^i$ 用 $[x^n]A(x)$ 表示 $A(x)$ 的 n 次项系数 a_n

实际运算时，通常只需要保留次数不超过 $n-1$ 的项进行计算 $(\%x^n)$

有标号和无标号

3个点的无标号简单无向图有4种

3个点的有标号简单无向图有8种

普通生成函数(OGF)

数列 $A_0, A_1, \dots$ 的普通生成函数定义为形式幂级数 $A(x) = \sum_{i \geq 0} A_i x^i$ 一个例子：

考虑 A_i, B_i 为两个背包，如何合并这两个背包 $C_k = \sum_{i+j=k} A_i B_j$ 对应于生成函数，
则有 $C(x) = A(x)B(x)$ 一个容量为 a 的物品的完全背包 $\sum_{i \geq 0} x^{ai}$ 写成封闭形式 $\frac{1}{1-x^a}$

指数生成函数(EGF)

数列 $A_0, A_1, \dots$ 的指数生成函数定义为形式幂级数 $A(x) = \sum_{i \geq 0} A_i \frac{x^i}{i!}$

EGF常用来解决带标号问题。

从 k 个标号里选出 i 个标号给 A_i 中的 i 个元素 $k-i$ 个标号给 B_j 中的 j 个元素。这样相当于对 C_k 中的元素编号。则 C_k 中带标号方案数

$$C_k = \sum_{i+j=k} \frac{k!}{i!(k-i)!} A_i B_j$$

$$\frac{C_k}{k!} = \sum_{i+j=k} \frac{A_i}{i!} \frac{B_j}{j!}$$

对应于生成函数，则有 $C(x) = A(x)B(x)$

指数与对数函数

$$\ln(1-A(x)) = -\sum_{i \geq 1} \frac{A(x)^i}{i} \quad \exp(A(x)) = \sum_{i \geq 0} \frac{A(x)^i}{i!}$$

例题

籀论文中的一道例题：完全背包计数

若干种不同的物品，体积为 i 的物品有 a_i 种，每种物品有无限个。求装满体积为 n 的背包的方案数 $\square(a_i, n \leq 10^5)$

体积为 i 的物品的生成函数 $\frac{1}{1-x^i}$ 将所有物品的生成函数相乘

$$F(x) = \prod_{i=1}^n \left(\frac{1}{1-x^i} \right)^{a_i}$$

$$\ln F(x) = \sum_{i=1}^n -a_i \ln(1-x^i)$$

对于右边来说，将 x^i 代入 $\ln(1-x)$ 的展开式得

$$\ln F(x) = \sum_{i=1}^n a_i \sum_{j \geq 1} \frac{x^{ij}}{j} = \sum_{T \geq 0} \sum_{i|T} \frac{a_i}{i} x^T$$

因为 T 最多到 n 根据调和级数，我们可以在 $O(n \log n)$ 内求出 $\ln F(x)$ 然后多项式 $\exp$ 一下，可以得到 $F(x)$

SDOI2017遗忘的集合

给定一个集合的完全背包 $f(1) \sim f(n)$ 求这个集合。集合中元素最大为 n

用 a_i 表示 i 这个元素是否为集合中的元素。

则整个集合中元素的生成函数 $F(x) = \prod_{i=1}^n \left(\frac{1}{1-x^i} \right)^{a_i}$ 与上一题同样可得 $\ln F(x) = \sum_{i=1}^n a_i \sum_{j \geq 1} \frac{x^{ij}}{j} = \sum_{T \geq 0} \sum_{i|T} \frac{a_i}{i} x^T$

$$n[x^n] \ln F(x) = \sum_{i|n} a_i$$

莫比乌斯反演一下就可以得到 a_i

付公主的背包

和完全背包计数很像

第一类斯特林数 · 行

S_n^m 表示把 n 个不同的元素构成 m 个圆排列的方案数（不能有空的环）

$S_n(x)$ 表示第 n 列斯特林数的生成函数 $S_n(x) = \sum_{i=0}^{\infty} S_n^i x^i$

$$S_n^i = S_{n-1}^{i-1} + (n-1) S_{n-1}^i$$

$$S_n(x) = x S_{n-1}(x) + (n-1) S_{n-1}(x) = (x+n-1) S_{n-1}(x) = \prod_{i=0}^{n-1} (x+i)$$

可以分治FFT在 $O(n \log^2 n)$ 的复杂度内求出，大概也有 $O(n \log n)$ 的求法。

第一类斯特林数 · 列

大概是采用了Zory的做法

考虑 n 个有标号元素组成的环的方案数 $(n-1)!$

考虑取一个带标号环的方案数，可以表示为EGF $\sum_{i \geq 1} (i-1)! \frac{x^i}{i!}$ 依次取 m 个环，

方案数 $\frac{1}{m!} \sum_{i \geq 1} (i-1)! \frac{x^i}{i!} \bigg)^m$

$S_n^m = n! [x^n] F(x)$

第二类斯特林数 · 行

S_n^m 表示把 n 个不同的元素分到 m 个相同的集合中（不能有空集）的方案数

把 m 个集合看成不同的，枚举几个集合没放，得到容斥的式子后 $\frac{1}{m!} \sum_{i=0}^m C_m^i (-1)^{m-i} i^n$

$S_n^m = \frac{1}{m!} \sum_{i=0}^m C_m^i (-1)^{m-i} i^n$
 $= \sum_{i=0}^m \frac{(-1)^{m-i} i^n}{(m-i)!} = \sum_{i=0}^m \frac{(-1)^i}{i!} \frac{(m-i)^n}{(m-i)!}$
 可以让 $\sum_{i=0}^m \frac{(-1)^i}{i!}$ 和 $\sum_{i=0}^m \frac{(m-i)^n}{(m-i)!}$ 做卷积

第二类斯特林数 · 列

大概是采用了 **Great Influence** 的做法

$S_m(x)$ 表示第 m 列斯特林数的生成函数 $S_m(x) = \sum_{i=0}^{\infty} S_i^m x^i$

$S_i^m = S_{i-1}^{m-1} + m S_{i-1}^m$ 可以得到 $S_m(x) = x S_{m-1}(x) + x m S_m(x)$

$S_m(x) = \frac{x}{1-mx} S_{m-1}(x) \Rightarrow \frac{S_m(x)}{x^m} = \prod_{i=1}^m (1-ix)$

$\prod_{i=1}^m (1-ix)$ 可以分治 FFT 在 $O(n \log^2 n)$ 的复杂度内求出，大概也有 $O(n \log n)$ 的求法。

2020牛客寒假算法基础集训营4 二维跑步

向左有2种跑法，向右有3种跑法，也可以原地不动。

如果用 $F[i][j]$ 表示跑了 i 步，当前在位置 j 的方案数 $F[i][j] = 3F[i-1][j-1] + F[i-1][j] + 2F[i-1][j+1]$ 令生成函数

$F_i(x) = \sum F[i][j] x^j$ 那么有 $F_n(x) = 3x F_{n-1}(x) + F_{n-1}(x) + \frac{2}{x} F_{n-1}(x)$

$= (3x+1+\frac{2}{x}) F_{n-1}(x) = (3x^2+x+2) F_{n-1}(x)$ 为了方便计算 $x^n F_n(x) = (3x^2+x+2)^n$ 对于右边，可以先求所有单位根 w 的 $(3w^2+w+2)^n$ 然后插值得到 $x^n F_n(x)$

复杂度 $O(n \log n)$ 标算用了更快的做法。

指数生成函数

POJ Blocks

用 RGBY 四种颜色涂满长度为 n 的序列，RG 这两种颜色需要出现偶数次。

这是一个带标号组合问题，标号可以理解为这个某个颜色的位置。

考虑 R 涂满长度为 $0 \sim n$ 的序列的方案数为 $1, 0, 1, 0, 1, 0, 1, 0, 1, \dots$

那么 R 的指数生成函数 $\sum_{i \geq 0} \frac{x^{2i}}{(2i)!} = \frac{e^x + e^{-x}}{2}$ 考虑 B 涂满长度为 $0 \sim n$ 的序列的方案数为 $1, 1, 1, 1, 1, \dots$

那么 R 的指数生成函数 $\sum_{i \geq 0} \frac{x^i}{i!} = e^x$ 将四种颜色的生成函数乘起来

$F(x) = \left(\frac{e^x + e^{-x}}{2} \right)^2 e^{2x} = \frac{1}{4} (e^{2x} + 1)^2 = \frac{1}{4} (e^{4x} + 2e^{2x} + 1)$

$= \frac{1}{4} + \frac{1}{2} \sum_{i \geq 0} \frac{4^i x^{2i}}{(2i)!} + \frac{1}{4} \sum_{i \geq 0} \frac{2^i x^i}{i!}$ $n \geq 1$ 时， $x^n [x^n] F(x) = 4^{n-1} + 2^{n-1}$ 可以用快速幂求

出。

或者 $f[i][j](0 \leq j \leq 3)$ 表示 dp 到第 i 个位置 RG 分别放了奇数次/偶数次得方案数，可以矩阵快速幂优化。

HUD 排列组合

有 n 种物品，并且知道每种物品的数量 a_i ，要求从中选出 m 件物品的排列数。例如有两种物品 A, B 并且数量都是 1，从中选 2 件物品，则排列有“AB”，“BA”两种 $n, m (1 \leq m, n \leq 10)$

这是一个带标号组合问题，标号可以理解为某个物品的位置。

第 i 个物品的 EGF $\sum_{i=0}^{a_i} \frac{1}{i!} x^i$ 将 n 个物品的 EGF 乘起来得 $F(x)$ 则答案为 $m! [x^m] F(x)$

luoguP4841 [集训队作业2013]城市规划

题意

求 n 个点的简单有标号无向连通图数目模 1004535809 。

题解

设 $f(n)$ 为点数为 n 的无向连通图的数量 $g(n)$ 为点数为 n 的无向图的数量。
那么有

$$g(n) = \sum_{i=1}^n \binom{n-1}{i-1} f(i) g(n-i)$$

，可以看成枚举一号节点所在连通块大小。

代入 $g(n) = 2^{\binom{n}{2}}$

$$2^{\binom{n}{2}} = \sum_{i=1}^n \binom{n-1}{i-1} f(i) 2^{\binom{n-i}{2}}$$

$$\frac{2^{\binom{n}{2}}}{(n-1)!} = \sum_{i=1}^n \frac{f(i)}{(i-1)!} \frac{2^{\binom{n-i}{2}}}{(n-i)!}$$

这是一个卷积，于是定义

$$F(x) = \sum_{n=1}^{+\infty} \frac{f(n)}{(n-1)!} x^n$$

$$G(x) = \sum_{n=0}^{+\infty} \frac{2^{\binom{n}{2}}}{n!} x^n$$

$$H(x) = \sum_{n=1}^{+\infty} \frac{2^{\binom{n}{2}}}{(n-1)!} x^n$$

则

$$F = H * G^{-1} \pmod{x^{n+1}}$$

CF438E The Child and Binary Tree

题意

一棵树的权值为其点权和。点权在正整数集合 $\{c_1, \dots, c_n\}$ 中，求不同的有根二叉树的权值和模 998244353。

题解

设 f_i 表示点权和为 i 的二叉树个数， g_i 表示权值 i 是否包含在 $\{c_1, \dots, c_n\}$ 中。 f_i 可由枚举根的点权和左右子树得到

$$f_0 = 1$$

$$f_n = \sum_{i=1}^n g_i \sum_{j=1}^{n-i} f_j f_{n-i-j} \quad (n > 0)$$

令 F 表示 f 的生成函数， G 表示 g 的生成函数，那么

$$F = G * F^2 + 1$$

解得

$$F = \frac{1 \pm \sqrt{1 - 4G}}{2G}$$

讨论

$$\text{取正号} \quad \lim_{x \rightarrow 0} F(x) = \infty \quad (\text{舍去})$$

$$\text{取负号} \quad \lim_{x \rightarrow 0} F(x) = 1 \quad (\text{满足})$$

于是得

$$F = \frac{1 - \sqrt{1 - 4G}}{2G}$$

$$F = \frac{2}{1 + \sqrt{1 - 4G}}$$

luoguP4451 [国家集训队]整数的lqp拆分

题意

F_n 表示斐波那契数列第 n 项。对给定 n 拆分指 $m > 0, a_1, a_2, \dots, a_m > 0$ ，且 $a_1 + a_2 + \dots + a_m = n$ 的有序集合，拆分的权值为 $F_{a_1} F_{a_2} \dots F_{a_m}$ ，求所有拆分权值之和模 1000000007。

题解

斐波那契数列的生成函数为

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

拆分为 k 个数，生成函数为 $F(x)^k$ ，于是答案的生成函数为

$$G(x)=\sum_{i=0}^{\infty} F(x)^i$$

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

分母 $\frac{1}{1-2x-x^2}$ 写成递推式

$$a_n=2a_{n-1}+a_{n-2}$$

乘上分子

$$a_n=a_n-a_{n-1}-a_{n-2}=a_{n-1}$$

解递推特征方程，得

$$x_1=-\sqrt{2}+1, x_2=\sqrt{2}+1$$

设通项公式为

$$a_n=c_1(-\sqrt{2}+1)^n+c_2(\sqrt{2}+1)^n$$

代入 $n=0, n=1$ 解得

$$c_1=\frac{\sqrt{2}-1}{2\sqrt{2}}, c_2=\frac{\sqrt{2}+1}{2\sqrt{2}}$$

$\sqrt{2}$ 在模 10^9+7 下有二次剩余 (59713600) ，所以最终答案

$$a_n \equiv 485071604 \times 940286408^n + 514928404 \times 59713601^n \pmod{10^9+7}$$

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:looking_up_at_the_starry_sky:%E7%94%9F%E6%88%90%E5%87%BD%E6%95%B0&rev=1590928155

Last update: 2020/05/31 20:29