生成函数入门

发表于 2022-01-12 | 更新于 2022-02-07 | 分类于 OI | Valine： 0

生成函数是解决计数问题的一个非常有效的手段，本文介绍了生成函数的定义以及基础运用。

← 前置知识：多项式基础操作 → 下一篇：斯特林数学习笔记

形式幂级数

形式幂级数 $\sum_i a_i x^i$，可以理解为项数无穷的多项式。

生成函数是形式幂级数，若某项以后的所有系数全为 0，就可以当做多项式处理。

常见形式幂级数封闭形式转化

1. $\sum\limits_{i=0}^{\infty} x^i = \frac{1}{1-x}$（类比等比数列）
2. $\sum\limits_{i=0}^{\infty} \binom{i+k}{k} x^i = \frac{1}{(1-x)^{k+1}}$（做 $k+1$ 次前缀和，每做一次相当于乘上 $\frac{1}{1-x}$）
   - 另：$\sum\limits_{i=0}^{\infty} \binom{i}{k} x^i = \sum\limits_{i=k}^{\infty} \binom{i}{k} x^i = \sum\limits_{i=0}^{\infty} \binom{i+k}{k} x^{i+k}$，上下都带 i 的可以通过上式拆过来。
3. $\sum\limits_{i=0}^{\infty} \binom{k}{i} x^i = (x+1)^k$（二项式定理）
4. $\sum\limits_{i=0}^{\infty} \frac{x^i}{i} = -\ln(1-x)$，$\sum\limits_{i=0}^{\infty} \frac{x^i}{i!} = \exp(x)$，$\sum\limits_{i=0}^{\infty} \frac{x^{2i}}{(2i)!} = \frac{\exp(x)-\exp(-x)}{2}$

普通生成函数（OGF）

定义

定义序列 f_i 的 OGF 为 $F(x) = \sum_i f_i x^i$，其中 f_i 表示规模为 i （由**互不区分的** i 个元素组成）的特定组合对象的个数。

记 $[x^i]F(x)$ 为 $F(x)$ 的 i 次项，在这里就是 f_i。这两个东西虽然在这里一样，但是实际上是不同的，在 EGF 时会提到。

普通生成函数一般解决的是**无标号**组合对象的计数问题。

运算

加法

- 组合意义：h_i 这个组合对象可以是一个规模为 i 的 f–对象，也可以是一个规模为 i 的 g–对象。
- 公式：$h_i = f_i + g_i$，即 $H(x) = \sum\limits_{i=0}^{\infty} (f_i + g_i) x^i$。
- 计算方式（前 n 项）：直接对位相加，时间复杂度 $O(n)$。

乘法

- 组合意义：h_i 是由两部分组成的**有序** pair 数量，第一部分是 f–对象，第二部分是 g–对象，两部分规模和为 i。
- 公式：$h_i=\sum\limits_{j=0}^{i} f_j g_{i-j}$，即 $H(x)=\sum\limits_{i=0}^{\infty}\left(\sum\limits_{j=0}^{i} f_j g_{i-j}\right)x^i$。（由于无标号，所以并没有系数）
- 计算方式（前 n 项）：FFT 或 NTT 做卷积，时间复杂度 $O(n\log n)$。

> **例：P2000 拯救世界**
> 太长，自己看。

Sol：写出 10 种石头的生成函数的封闭形式，然后乘在一起得到答案。

比如第一种，必须是 6 的倍数，也就是 $1+x^6+x^{12}+\cdots=\frac{1}{1-x^6}$，其他具体的推导过程见题解区 =w=

最后总之就是乘起来得到 $\frac{1}{(1-x)^5}$，套用公式得到第 n 项系数为 $\binom{4}{n+4}$。但是这就引出了一个问题：如果最后推出来的生成函数形如 $F(x)=\frac{P(x)}{Q(x)}$，$Q(x)$ 又是奇怪的东西，复杂度又不允许求逆，该如何处理?

- 递推式

移项一下，$P(x)=F(x)Q(x)$，提取第 n 项 $[x^n]P(x)=\sum\limits_{i=0}^{n}[x^i]F(x)\cdot[x^{n-i}]Q(x)$，将 $[x^n]F(x)$ 移至左边得到 $[x^n]F(x)q_0=[x^n]P(x)-\sum\limits_{i=0}^{n-1}[x^i]F(x)\cdot[x^{n-i}]Q(x)$。

有常系数线性齐次递推的做法，但是我不会，先鸽 QAQ

生成序列与集合

生成序列

- 组合意义：g_i 是由任意个规模之和为 i 的 f–对象排成的**序列**个数。（注意此时 $f_0=0$ 是必需的，否则会有无数个这样的序列）

- 公式

先考虑由 k 个规模之和为 i 的 f–对象组成的序列个数 $[x^i]P_k$，其生成函数 $P_k(x)=F(x)^k$，而显然 h_i 可以是任意个，也就是 $[x^i]G(x)=\sum\limits_{k=1}^{i}[x^i]P_k(x)$。故能导出公式：

$$G(x)=\sum_{k=0}^{\infty}P_k(x)=\sum_{k=0}^{\infty}F(x)^k=\frac{1}{1-F(x)}$$

- 计算方式（前 n 项）

使用多项式求逆，时间复杂度 $O(n\log n)$。

> **例：有序划分**
> 将正整数 n 划分成有序正整数的方案数。

Sol：划分为有序正整数就相当于生成任意长的正整数序列使得和为 n，而划分为一个正整数的生成函数

为 $F(x)=x+x^2+\cdots=\frac{x}{1-x}$，故原问题生成函数 $G(x)=\frac{1}{1-F(x)}=\frac{1-x}{1-2x}$。

生成集合

- 组合意义：g_i 是由任意个规模之和为 i 的 f–对象组成的**集合**个数。
- 公式

既然是集合，就枚举所有可能的 f–对象，以及它在集合中出现多少次。对于规模为 i 的**一种**对象，它可以在集合里出现 0 次或者多次，且都只有一种方案（因为生成的是集合），所对应的生成函数即为 $1+x^i+x^{2i}+\cdots=\frac{1}{1-x^i}$。而规模为 i 的 f–对象有 f_i 种，我对每种对象要依次决策在最终集合中放多少个，所对应的生成函数即为 $\left(\frac{1}{1-x^i}\right)^{f_i}$。故有：

$$G(x)=\prod_{i=1}^{\infty}\left(\frac{1}{1-x^i}\right)^{f_i}$$

- 计算方式（前 n 项）

显然上式这个求积的形式并不是给人算的，考虑取对变成求和：

$$\begin{aligned}\ln G(x)&=\sum_{i=1}^{\infty}f_i\ln\left(\frac{1}{1-x^i}\right)\\&=\sum_{i=1}^{\infty}f_i\cdot-\ln\left(1-x^i\right)\\&=\sum_{i=1}^{\infty}f_i\sum_{j=1}^{\infty}\frac{(x^i)^j}{j}\end{aligned}$$

此时交换枚举顺序，试图凑出 $F(x)$ 的形式：

$$\begin{aligned}LHS&=\sum_{j=1}^{\infty}\frac{1}{j}\sum_{i=1}^{\infty}f_i(x^j)^i\\&=\sum_{j=1}^{\infty}\frac{F(x^j)}{j}\end{aligned}$$

再 exp 回去，于是我们得到了一个相当简洁的形式：

$$G(x)=\exp\left(\sum_{i=1}^{\infty}\frac{F(x^i)}{i}\right)$$

括号内的用调和级数相加，然后做一次 exp 得到 $G(x)$，时间复杂度 $O(n\log n)$。

> **例：无序划分**
> 将正整数 n 划分成无序正整数的方案数。

Sol：划分为无序正整数就相当于生成任意大小的正整数多重集（或者非严格升序正整数序列）使其和为 n，而划分为一个正整数的生成函数为 $F(x)=x+x^2+\cdots=\frac{x}{1-x}$，故原问题生成函数 $G(x)=\exp\left(\sum_{i=1}^{\infty}\frac{F(x^i)}{i}\right)$。

指数生成函数（EGF）

定义

定义序列 f_i 的 EGF 为 $F(x)=\sum_i f_i\frac{x^i}{i!}$，其中 f_i 表示在 i 个**互相区分**的元素间可以建立的不同的结构（e.g. 序列）数。

记 $[x^i]F(x)$ 为 $F(x)$ 的 i 次项，在这里就是 $\frac{f_i}{i!}$。$[x^i]F(x)$ 和 f_i 的不同之处就在于，f_i 表示组合方案数，但实际上维护多项式的系数时维护的是 $[x^i]F(x)$，在这里就产生了不同。

指数生成函数一般解决的是**有标号**组合对象的计数问题。至于为什么这种形式是有标号的，在后面会解释，暂时可以只记住定义。

运算

加法

- 组合意义：h_i 表示我可以在 i 个对象间建立 f–结构或者是 g–结构。
- 公式：$h_i=f_i+g_i$，即 $H(x)=\sum\limits_{i=0}^{\infty}(f_i+g_i)\frac{x^i}{i!}$。
- 计算方式（前 n 项）：直接对位相加，时间复杂度 $O(n)$。

乘法

- 组合意义：h_i 表示有 i 个对象，我从中挑选出 j 个位置建立 f–结构，$(i-j)$ 个建立 g–结构的方案数。
- 公式

i 选 j，显然系数要乘上一个组合数：$h_i=\sum\limits_{j=0}^{i}\binom{i}{j}f_jg_{i-j}$，即 $H(x)=\sum\limits_{i=0}^{\infty}\left(\sum\limits_{j=0}^{i}\binom{i}{j}f_jg_{i-j}\right)\frac{x^i}{i!}$。

这里解释一下为什么 EGF 这里除以一个 $i!$ 就能解决有标号问题：把上式写成多项式中 i 次项的形式，就是 $i![x^i]H(x)=\sum\limits_{j=0}^{i}\binom{i}{j}\cdot j!\cdot[x^j]F(x)\cdot(i-j)!\cdot[x^{i-j}]G(x)$，发现组合数恰好消掉了，写成 $[x^i]H(x)=\sum\limits_{j=0}^{i}[x^j]F(x)\cdot[x^{i-j}]G(x)$ 就是卷积了。

- 计算方式（前 n 项）：FFT 或 NTT 做卷积，时间复杂度 $O(n\log n)$。

> **例：P5339 [TJOI2019]唱、跳、rap和篮球**
> 若一个序列内不存在一个连续四个数，从左至右分别是 $(1,2,3,4)$，就称之为合法的。
>
> 你手上的 $1,2,3,4$ 的个数分别有 a,b,c,d 个，需要排成一个长度为 n 的序列（$a+b+c+d\geq n$），问方案数。
>
> $n\leq 1000$，对 998244353 取模。

Sol：首先令 $f(i)$ 表示序列中恰好 i 组「不合法四元组」的方案数，$g(i)$ 表示序列中钦定了 i 组「不合法四元组」的方案数，我们求出 g 后二项式反演即可得到 f，问题是如何求 $g(i)$。

观察可得，构造一个 $g(i)$ 分为两个过程：

1. 在长度为 n 的序列中选出 i 个「不合法四元组」。

可以看成，从长度为 $n-3i$ 的序列中，挑选 i 个位置变成「不合法四元组」，方案数 $\binom{n-3i}{i}$。

2. 在剩下四个 $n-4i$ 个位置中随便填，但使用的 $1,2,3,4$ 数量不能超过 $a-i,b-i,c-i,d-i$ 的不同序列数。

以 $a-i$ 个 1 为例，它能排成 1 种长度不超过 $a-i$ 的全 1 序列，EGF 为 $\sum\limits_{j=0}^{a-i}\frac{x^j}{j!}$。而我们要把这四个序列”合成“为一个序列，也就是这四个 EGF 卷起来之后得到的 $H(x)$ 即为原问题的生成函数，答案即 $[x^n]H(x)\cdot n!$，时间复杂度 $O(n^2\log n)$。

> **例：P5219 无聊的水题 I**
> 求 n 个节点、节点最大度数为 m 的有标号树个数。
>
> $n,m\le 5\times 10^4$，对 998244353 取模。

Sol：x 的度数等于其在 Prufer 序列中出现次数 $+1$，转化为长度为 $n-2$，元素出现次数的最大值恰为 $m-1$ 的序列个数，进一步转化为最大值不超过 $m-1$ 的减去不超过 $m-2$ 的。

所以考虑不超过 m 的情况如何解决：考虑每一个值，其排成长度 $\le m$ 的序列有 1 种方案，否则 0 种方案。使用 EGF 表示即为 $F(x)=\sum\limits_{i=0}^{m}\frac{x^i}{i!}$。由于有 n 种值，故答案的 EGF 为 $F(x)^n$，使用多项式快速幂即可。

生成序列与集合

EGF 与 OGF 生成序列/集合的方式的不同点在于：

生成序列

- 组合意义：g_i 是将 i 个可区分元素，**有顺序地**划分成规模之和为 i 的任意组，并在每组内建立一个 f–结构的方案数。（也就是将一个序列染成若干种颜色，每一种颜色建立一个 f–结构）
- 公式：同 OGF，$G(x)=\frac{1}{1-F(x)}$，多项式求逆 $O(n\log n)$ 解决。

> **例：P5162 WD与积木**
> n 块有标号积木，每个积木大小随机，堆积木时会把大小相同的放在一层，并且层与层之间从大到小排序。
>
> 两种堆法不同当且仅当某个积木在两种堆法中处于不同的层中，所有不同的堆法等概率出现，问期望层数 $\bmod 998244353$。T 组询问，$T,n\le 10^5$。

Sol：由于是有标号问题，使用 EGF。考虑求出总方案数与所有方案长度之和。

首先考虑一层的方案 $F(x)=\sum\limits_{x\ge 1}\frac{x^i}{i!}=\exp(x)-1$，则总方案数是生成一个以层为元素的序列（因为层是有序的），生成函数即为 $\frac{1}{1-F(x)}$。

再算长度之和，枚举层数求方案。堆 i 层方案为 $F(x)^i$，故结果为 $\sum\limits_{i=1}^{\infty}i\cdot F(x)^i=\frac{F(x)}{(1-F(x))^2}$。原问题

即提取第 n 项系数相除。

生成集合

- 组合意义：g_i 是将 i 个可区分元素划分成规模之和为 i 的任意组并在每组之间建立 f-结构，但组与组之间不区分顺序的方案数。
- 公式：由于可以互相区分，所以方案数即为同样规模的序列方案数除以 $k!$，即 $G(x)=\sum\limits_{k=0}^{\infty}\frac{F(x)^k}{k!}=\exp(F(x))$。

> **例：P4841 [集训队作业2013]城市规划**
> 带标号无向连通图计数，$n\le 130000$。

Sol：一个带标号无向图的方案数，显然可以看成：将其分为若干组，在每组中建立一个无向联通图，且组与组之间不区分。而 n 个点带标号无向图的方案数显然是 $h_n=2^{\binom{n}{2}}$，其 EGF 为 $H(x)=\sum\limits_{i=1}^{\infty}h_i\frac{x^i}{i!}$。而令原问题生成函数为 $F(x)$，则有 $H(x)=\exp(F(x))$，故 $F(x)=\ln(H(x))$。