生成函数与多项式

广州二中 马耀华

生成函数

- 无穷数列$< a_0, a_1, a_2, \ldots >$的一般生成函数（OGF）是定义在形式幂级数环上的形式幂级数$A(z) = \sum_{i\geq 0} a_i z^i$。
- 它对应的指数生成函数（EGF）是$\hat{A}(z) = \sum_{i\geq 0} a_i \frac{z^i}{i!}$。
- 一些常用的生成函数:
- $\frac{1}{1-z} \quad \sum_{i\geq 0} z^i \quad < 1, 1, 1, 1, \ldots >$
- $\ln \frac{1}{1-z} \quad \sum_{i>0} \frac{1}{i} z^i < 0, 1, \frac{1}{2}, \frac{1}{3}, \ldots >$
- $e^z \quad \sum_{i\geq 0} \frac{z^i}{i!} \quad < 1, 1, \frac{1}{2}, \frac{1}{6}, \ldots >$
- $(1+z)^c \quad \sum_{i\geq 0} \binom{c}{i} z^i \quad < 1, c, \binom{c}{2}, \binom{c}{3}, \ldots >$
- $\frac{1}{(1-z)^c} \quad \sum_{i\geq 0} \binom{c+i-1}{i} z^i \quad < 1, c, \binom{c+1}{2}, \binom{c+2}{3}, \ldots >$

简单运算

- 两个生成函数的加法对应两个数列对应项相加:

$$< a_0 + b_0, a_1 + b_1, a_2 + b_2, a_3 + b_3, \ldots >$$

$$\left(\sum_{i\geq 0} a_i z^i\right) + \left(\sum_{i\geq 0} b_i z^i\right)$$

- 乘法对应两个数列卷积（可能需要FFT/NTT）：

$$< a_0b_0, a_0b_1 + a_1b_0, a_0b_2 + a_1b_1 + a_2b_0, \ldots >$$

$$\left(\sum_{i\geq 0} a_i z^i\right) * \left(\sum_{i\geq 0} b_i z^i\right) \text{ OGF}$$

$$< a_0b_0, \binom{1}{0}a_0b_1 + \binom{1}{1}a_1b_0, \binom{2}{0}a_0b_2 + \binom{2}{1}a_1b_1 + \binom{2}{2}a_2b_0, \ldots >$$

$$\left(\sum_{i\geq 0} a_i \frac{z^i}{i!}\right) * \left(\sum_{i\geq 0} b_i \frac{z^i}{i!}\right) \text{ EGF}$$

- 分治FFT（半在线卷积？）

简单运算

- 数列左移和右移:

$$< 0, 0, ...(\times m), a_0, a_1, ... >$$

$$z^m(\sum_{i\geq 0} a_i z^i)$$

$$< a_m, a_{m+1}, ... >$$

$$\frac{(\sum_{i\geq 0} a_i z^i) - (\sum_{i=0}^{m-1} a_i z^i)}{z^m}$$

- 求导

$$< a_1, 2a_2, 3a_3, ... >$$

$$(\sum_{i\geq 0} a_i z^i)'$$

- 积分

$$< 0, a_0, \frac{a_1}{2}, \frac{a_2}{3}, ... >$$

$$\int_0^z (\sum_{i\geq 0} a_i t^i)\mathrm{d}t$$

牛顿迭代

- 考虑一个方程 $F(x)=0$ ，这里F和x都是关于z的形式幂级数，给定F要解出x。
- 一般来说x是一个无穷级数，我们会截断前面一部分，只需求出 $x \mod z^n$ 。

牛顿迭代

- 考虑一个方程 $F(x)=0$，这里F和x都是关于z的形式幂级数，给定F要解出x。
- 一般来说x是一个无穷级数，我们会截断前面一部分，只需求出 $x \mod z^n$。
- 若我们已经求出了 $x_0 = x \mod z^n$，试图求出 $x \mod z^{2n}$，考虑在 $x=x_0$ 处展开，有
$$F(x)=F(x_0)+F'(x_0)(x-x_0)+F''(x_0)\frac{(x-x_0)^2}{2}+...=0$$
- 在 $\mathrm{mod}\ z^{2n}$ 意义下考虑，显然有 $x\equiv x_0-\frac{F(x_0)}{F'(x_0)} \pmod{z^{2n}}$。
- 倍增即可，一般复杂度跟计算单次函数复合相同。

求逆

- 给定生成函数$F(z)$，求$G(z)$使得$F(z)G(z) \equiv 1 \pmod{z^n}$。（常数项不为0）。

求逆

- 给定生成函数$F(z)$，求$G(z)$使得$F(z)G(z) \equiv 1 \pmod{z^n}$。（常数项不为0）。
- 考虑方程$\frac{1}{x} - F(z) = 0$，牛顿迭代有
$$x \equiv x_0(2 - x_0F(z)) \pmod{z^{2n}}$$
- 直接倍增复杂度为$\mathcal{O}(n \log n)$。

取模

- 求出 $A(z) \mod B(z) \quad (deg(A(z)) = n, deg(B(z)) = m)$

取模

- 求出 $A(z) \mod B(z) \quad (deg(A(z)) = n, deg(B(z)) = m)$
- 考虑对$F(z)$ 翻转系数得到的$F^R(z) = z^{deg(F(z))}F(\frac{1}{z})$
- 做带余除法 $A(z) = B(z)F(z) + G(z) \quad (deg(G(z)) < n - m)$
- ，两边做变换有 $A^R(z) = B^R(z)F^R(z) + z^{n-deg(G(z))}G^R(z)$ 。
- 考虑两边同时 $\mathrm{mod}\ z^{n-m+1}$，有 $A^R(z) \equiv B^R(z)F^R(z) \pmod{z^{n-m+1}}$
- ，于是 $F^R(z) \equiv A^R(z) \cdot (B^R(z))^{-1} \pmod{z^{n-m+1}}$
- 求出了$F(z)$ 也就容易求出$G(z)$。
- 复杂度同样为$\mathcal{O}(n \log n)$ 。

对数与指数

- 给定 $F(z)$，求 $\ln F(z) \pmod{z^n}$（常数项为1）。
- 给定 $G(z)$，求 $e^{G(z)} \pmod{z^n}$ （常数项为0）。

对数与指数

- 给定 $F(z)$，求 $\ln F(z) \pmod{z^n}$（常数项为1）。
- 给定 $G(z)$，求 $e^{G(z)} \pmod{z^n}$ （常数项为0）。

- 对数: $\ln F(z) = \int_0^z \frac{F'(t)}{F(t)} \mathrm{d}t$ ，直接按式子计算。
- 复杂度为 $\mathcal{O}(n \log n)$ 。
- 指数：牛顿迭代，方程为 $\ln x - G(z) = 0$ 。
- 迭代可得 $x \equiv x_0(1 - \ln x_0 + G(z)) \pmod{z^{2n}}$。
- 复杂度同样为 $\mathcal{O}(n \log n)$。
- 经典用途：图个数EGF与连通图个数EGF的互相转化。

求幂

- 求 $F^k(z) \pmod{z^n}$，k可以是任意实数。

求幂

- 求 $F^k(z) \pmod{z^n}$，k可以是任意实数。
- 当常数项为1时，由$F^k(z) = \exp(\ln F(z) * k)$可以快速计算。
- 当常数项不为1时，若常数项为0可以平移一下，变成常数项不为0，再求个逆元即可。
- 时间复杂度还是$\mathcal{O}(n \log n)$。

链式反应

- 解一个一阶微分方程 $F'(z) = \frac{1}{2}A(z)F^2(z) + 1$
- 只需要算 $F(z) \mod z^n$

多点求值

- 给定一个n次多项式$F(z)$，分别求 $F(x_1), F(x_2), ..., F(x_m)$ 。

多点求值

- 给定一个n次多项式$F(z)$，分别求 $F(x_1), F(x_2), ..., F(x_m)$ 。
- 经典做法：$F(x_i) = F(z) \mod (z - x_i)$
- 分治，预先求出[l,r]对应的多项式 $\prod_{i=l}^{r}(z - x_i)$，然后从上往下递归，给[l,r]传入一个$F_{[l,r]}(z)$，对于[l,mid]传一个
- $F_{[l,r]}(z) \mod \prod_{i=l}^{mid}(z - x_i)$ [mid+1,r] 传一个
- $F_{[l,r]}(z) \mod \prod_{i=mid+1}^{r}(z - x_i)$ ，叶子节点剩下的一定是一个常数，复杂度为$\mathcal{O}(n \log n + m \log^2 n)$。
- 常数非常大，但目前在某些问题上有不可替代性（如范德蒙德行列式求值）。

多点求值

- 给定一个n次多项式$F(z)$，分别求 $F(x_1),F(x_2),...,F(x_m)$ 。
- 牛逼做法：令 $MULT(F(z),G(z))=\sum_{i\geq 0}(\sum_{j\geq 0} f_{i+j}g_j)z^i$
- 则$F(x_i)=[z^0]MULT(F(z),\frac{1}{1-x_iz})$。且有 $MULT(F(z),G(z)H(z))=MULT(MULT(F(z),G(z)),H(z))$
- 于是类似分治。根节点处传入一个 $MULT(F(z),\frac{1}{\prod_i(1-x_iz)})$，每次向[l,mid]传入$MULT(F_{[l,r]}(z),\prod_{i=mid+1}^{r}(1-x_iz))$，向[mid+1,r]传入$MULT(F_{[l,r]}(z),\prod_{i=l}^{mid}(1-x_iz))$ 。
- 常数小，传说中的1s5e5多点求值。

一个应用

为了找到使人脱离 Chaos Child 症候群的方法，Takuru 的大脑需要被扫描，而扫描时 Takuru 需要通过打游戏来保持大脑的活动。

他在游戏里和一个 ID 叫 KnightHart 的玩家一起刷着副本，打完之后获得了 n 件物品。Takuru 的等级一开始是 0，如果他使用了第 i 件物品，那么他的等级会有 p_i 的概率升一级，剩下 $1-p_i$ 的概率不变。如果 Takuru 的等级是k，那么他的攻击力就是 a_k。由于 KnightHart 是一名人生赢家，所以他只会拿走其中的一件物品，而 Takuru 会使用剩下的所有物品。

求对于每个 $1 \leqslant i \leqslant n$，如果 KnightHart 拿走了第 i 件物品，那么 Takuru 攻击力的期望是多少。

快速插值

- 给定n+1个点$(x_0, F(x_0)), (x_1, F(x_1)), ..., (x_n, F(x_n))$，求一个次数不超过n的多项式$F(z)$。

快速插值

- 给定n+1个点$(x_0,F(x_0)),(x_1,F(x_1)),...,(x_n,F(x_n))$，求一个次数不超过n的多项式$F(z)$。
- 考虑拉格朗日插值公式: $F(z)=\sum_{i=0}^{n}F(x_i)\prod_{j\neq i}\frac{z-x_j}{x_i-x_j}$
- 显然如果对每个i算出了$\prod_{j\neq i}(x_i-x_j)$，就可以分治FFT算出$F(z)$。
- 令$G(z)=\prod_{i=0}^{n}(z-x_i)$，由洛必达易知$G'(x_i)=\prod_{j\neq i}(x_i-x_j)$。
- 那么分治乘出$G(z)$后容易得到$G'(z)$，再多点求值即可。
- 最后再分治一下还原出$F(z)$，时间复杂度为$\mathcal{O}(n\log^2 n)$。

复合

- 给定 $F(z), G(z)$，求出$F(G(z)) \mod z^n$。（G常数项为0）

复合

- 给定 $F(z), G(z)$，求出$F(G(z)) \mod z^n$。（G常数项为0）
- 考虑直接展开式子，即为 $\sum_{i=0}^{n-1} f_i G^i(z) \pmod{z^n}$。
- 直接做复杂度为$\mathcal{O}(n^2 \log n)$。
- 稍微优化一下，令$M = \lceil \sqrt{n} \rceil$，求出
- $G(z), G^2(z), ..., G^M(z), G^{2M}(z), ..., G^{M^2}(z) \pmod{z^n}$，顺便把这些东西全部算一个2n次的DFT。然后还是按定义计算，对于某一项就可以变成两个已经算过DFT的东西的乘积了，单次可以$\mathcal{O}(n)$求出一项的DFT后的东西，最后加起来再IDFT回去就行了。
- 复杂度$\mathcal{O}(n^2 + n\sqrt{n} \log n)$，在OI范围内比$\mathcal{O}((n \log n)^{1.5})$ 做法快。

复合逆

- 给定常数项为0，一次项为1的多项式$F(z)$，求满足$G(F(z)) = z$
- 的多项式$G(z) \mod z^n$。注意由于多项式在复合运算下形成一个群, $G(F(z)) = z$意味着$F(G(z)) = z$。称F, G互为复合逆。

复合逆

- 给定常数项为0，一次项为1的多项式$F(z)$，求满足$G(F(z))=z$
- 的多项式$G(z) \mod z^n$。注意由于多项式在复合运算下形成一个群，$G(F(z))=z$意味着$F(G(z))=z$。称F,G互为复合逆。
- 考虑直接按定义牛顿迭代，解方程$F(x)-z=0$。
- 则$x \equiv x_0 - \frac{F(x_0)-z}{F'(x_0)} \pmod{z^{2n}}$，复杂度与计算复合$F(x)$相同。
- 当$F(x)$比较特殊或系数很稀疏时，也可以达到较优秀的复杂度。
- 如果仅需求出$[z^n]G(z)$，利用下面介绍的拉格朗日反演可以做到更优秀的复杂度。

拉格朗日反演

- 拉格朗日反演：若$F(x)=z$，则
$$[z^n]x=\frac{1}{n}[w^{-1}](\frac{1}{F(w)})^n$$
- 需要注意这里事实上是在分式环下讨论的（即使常数项为0的多项式也可逆），因此才会出现取-1次项。
- 更为实用的形式为$[z^n]x=\frac{1}{n}[w^{n-1}](\frac{w}{F(w)})^n$。
- 也有扩展形式$[z^n]H(x)=\frac{1}{n}[w^{-1}]H'(w)\cdot(\frac{1}{F(w)})^n$，或
- $[z^n]H(x)=\frac{1}{n}[w^{n-1}]H'(w)\cdot(\frac{w}{F(w)})^n$。
- 于是求复合逆单项可以$\mathcal{O}(n\log n)$。

证明

- 令 $G(F(z))=z$，有 $H(G(F(z)))=H(z)$ 。
- 令$P(z)=H(G(z))$，两边对z求导有$\sum_{i\geq 0}(i+1)p_{i+1}F^i(z)F'(z)=H'(z)$
- 同除$F^n(z)$，右边为$H'(z)\cdot(\frac{1}{F(z)})^n$，左边则是
- $\sum_{i\geq 0}(i+1)p_{i+1}F^{i-n}(z)F'(z)$。再同时取-1次项。
- 考虑对于$i\neq n-1$的项,$F^{i-n}(z)F'(z)=\frac{(F^{i-n+1}(z))'}{i-n+1}$，取-1次项后为0。
- $i=n-1$时, $[z^{-1}]F^{i-n}(z)F'(z)=1$，于是左边-1次项的值为$np_n$ 。
- 重新写一下就是上一页的形式了。

简单应用

- 拉格朗日反演相当重要，因为它是少数能够从生成函数封闭形式得到系数的方法。
- 例如n个点的二叉树计数，众所周知方案数为 $\frac{\binom{2n}{n}}{n+1}$ 。
- 用拉格朗日反演推的话，OGF为 $F(z) = z(1+F(z))^2$ ，故 $F(z)$
- 复合逆为 $G(z) = \frac{z}{(1+z)^2}$ 。直接套拉格朗日反演，有
- $[z^n]F(z) = \frac{1}{n}[w^{n-1}](\frac{w}{G(w)})^n = \frac{1}{n}[w^{n-1}](1+w)^{2n} = \frac{\binom{2n}{n}}{n+1}$ 。
- 类似对k叉树计数也能这么做。

特别应用

- 考虑在特殊的二元生成函数上做一个拉格朗日反演：
$$[z^n]\frac{1}{1-uF(z)} = \frac{1}{n}[w^{n-1}](\frac{1}{1-uw})' \cdot (\frac{w}{F^{-1}(w)})^n$$
- 这个形式相当有用，因为$(\frac{1}{1-uw})'$形式相当简单。
- 当$F^{-1}(w)$可以快速求得时，这个式子可以$\mathcal{O}(n\log n)$计算。
- 一个简单应用是求出 $[z^n]F(z), [z^n]F^2(z), ..., [z^n]F^n(z)$ 。
- 这相当于计算$[z^n]\frac{1}{1-uF(z)}$，于是当$F(z)$有复合逆时可以快速计算。
- 没有复合逆的话需要做一点变形，也是能算的。

[ZJOI2020] 抽卡

Bob 喜欢抽卡。

Bob 最近入坑了一款叫"主公连结"的抽卡游戏。游戏中共有 n 个不同的角色，编号为 $1 \sim n$。当 Bob 获得其中的编号连续的 k 张卡时，就可以组出理论最强阵容。

当期卡池中共有 m 张不同的卡，每次抽卡，Bob 都可以等概率随机获得一张卡池中的卡。如果 Bob 抽到了一张他已经拥有的卡，那么什么事都不会发生，等于 Bob 浪费了这次抽卡机会。Bob 是个谨慎的人，他想知道，如果他不停地抽卡直到抽到编号连续的 k 张卡时停止抽卡，期望需要抽多少轮。

常系数线性递推

- 给一个递推数列 $f_n = \sum_{i=1}^{m} a_i f_{n-i} \quad (n > m)$，快速求单项。
- 一个经典的做法是矩阵快速幂。
- 另一个经典做法是利用特征多项式和凯莱-哈密尔顿定理:
- $f(A) = 0$ 其中 $f(\lambda) = \det(\lambda I - A)$ 。这里只需要算出
- $\lambda^n \mod (\lambda^n - \sum_{i=1}^{m} a_i \lambda^{m-i})$，再乘上对应初始系数即可。
- 时间复杂度为 $\mathcal{O}(m^2 \log n)$ 或 $\mathcal{O}(m \log m \log n)$。
- 经常需要结合Berlekamp-Massey算法，预先求出特征多项式。
- 不是今天的重点就不展开了。

TRIPWAYS

大厨最近接受了来自某个著名的烹饪学校的教职。这份工作还没有正式开始，所以大厨打算利用剩下的时间好好地度个假。有 N 座城市（编号 $1 \sim N$），由 M 条道路相连。对每个合法的 i，第 i 座城市内有 L_i 个旅游景点。大厨现在在城市 1，他将会在城市 N 教书。在他度假的每一天，他会进行如下活动中的某一种：

- 走到一个编号比他目前所在城市要高的城市，要求这个城市与他目前所在的城市之间有道路连接。在假期结束的时候，大厨必须在城市 N。
- 访问一个他目前所在城市中的旅游景点。大厨可以（在不同的时间）重复访问同一个旅游景点。

大厨还没有决定度多久的假。他有 Q 个询问，由序列 $D_1, D_2, \ldots, D_Q$ 描述。对每个询问（也就是对每个 i，其中 $1 \leq i \leq Q$），他希望知道如果他的假期恰好长 D_i 天的话，不同的可能的度假计划的个数。由于这个数可能非常大，请你求出它模 $1,000,000,007$ 的值。

我们认为两个（持续时间相同的）度假计划不同，如果存在某一天使得大厨在这两个计划中做的事情不一样。访问两个不同的旅游景点也算不一样的事情。

TRIPWAYS

设$F_i(x)$表示到达点i的方案数关于时间的生成函数，显然有$F_1(x)=\frac{1}{1-L_1x}$，$F_i(x)=\frac{x\cdot\sum_{(j,i)\in E}F_j(x)}{1-L_ix}$ $(i>1)$ 。那么可以发现$F_N(x)$可以写成$\frac{P(x)}{Q(x)}$的形式，其中$deg(P(x))<N$，$Q(x)=\prod_{i=1}^{N}(1-L_ix)$。至于求出$P(x)$，可以考虑先DP出$F_N(x)$的前$N$项系数，乘上$Q(x)$即可，DP时间复杂度为$\mathcal{O}(NM)$。

现在每个询问即为给定k，求出$[x^k]F_N(x)$。如果直接用多项式取模加速常系数线性递推的经典算法，不仅单个询问复杂度为$O(N\log N\log k)$，且因为模数不是NTT模数，需要任意模数FFT，因此常数非常大，不能通过。

可以发现现在给定了$Q(x)$的因子分解，可以尝试得到复杂度更优秀的算法。

考虑给L_i排序，令去重后得到长度为d的数列p，其中p_i在L中出现了r_i次。根据有理生成函数的一般展开定理，我们可以将$F_N(x)$表示为$\sum_{i=1}^{d}\sum_{j=1}^{r_i}\frac{a_{i,j}}{(1-p_ix)^j}$，那么$[x^k]F_N(x)=\sum_{i=1}^{d}\sum_{j=1}^{r_i}a_{i,j}\cdot\binom{k+j-1}{j-1}\cdot {p_i}^k$。这样只需要快速幂就可以在$\mathcal{O}(N\log k)$的时间复杂度内处理单个询问。

现在难点在于得到这个分解，也即得到每个常数$a_{i,j}$。令$R_i(x)=\prod_{j\neq i}(1-p_jx)^{r_j}$，注意到通分后有$P(x)=\sum_{i=1}^{d}\sum_{j=1}^{r_i}(a_{i,j}\cdot R_i(x)\cdot(1-p_ix)^{r_i-j})$。那么有$P(\frac{1}{p_i})=a_{i,r_i}\cdot\prod_{j\neq i}(1-\frac{p_j}{p_i})^{r_j}$，于是容易得到$a_{i,r_i}$。得到$a_{i,r_i}$后从$P(x)$中减去对应项并整体除去一个$(1-p_ix)$即可将$r_i$减去1，继续计算即可。这样分解时间复杂度为$\mathcal{O}(N^2)$。

这里Entropy Increaser给出了一个复杂度更优秀的分解做法，也记录一下。我们令$G_i(x)=P(x)\bmod (1-p_ix)^{r_i}=((\sum_{j=1}^{r_i}a_{i,j}\cdot(1-p_ix)^{r_i-j})\cdot R_i(x))\bmod (1-p_ix)^{r_i}$，计算每个$G_i(x)$可以用类似多点求值的分治过程。注意到如果我们将$G_i(x)$和$R_i(x)$的基变换到$(1-p_ix)^k$下，也即令$G'_i(1-p_ix)=G_i(x)$，$R'_i(1-p_ix)=R_i(x)\bmod (1-p_ix)^{r_i}$，求出$G'_i(x)$和$R'_i(x)$的系数表示，这个可以二项式展开一下用卷积实现。那么$a_{i,j}=[x^{r_i-j}](G'_i(x)\cdot R'_i(x)^{-1})$，直接多项式求逆即可。这样分解时间复杂度为$\mathcal{O}(N\log^2 N)$。

总时间复杂度为$\mathcal{O}(NM+N^2+QN\log V)$或$\mathcal{O}(NM+N\log^2 N+QN\log V)$。

任意模数FFT

- 非NTT模数下的FFT：三模数NTT or 拆系数MTT
- 三模数NTT：选择三个比较大的NTT模数分别算，CRT合并。
- 缺点：常数很大，CRT的时候很麻烦
- 优点：在一些奇妙问题中不可替代（？）例：二项卷积
- 拆系数MTT：选择一个 $M = \lceil \sqrt{V} \rceil$，分开做几遍。
- 缺点：需要实数运算，有时候特殊问题不适用
- 优点：跑的快，比较好写。也可以增加代码长度来减少FFT次数，具体看论文。

一个白嫖的问题

- 给定一个n次多项式$F(z)$和一个k，q次询问不同的n：

$$\sum_{i=0}^{n} F(i)k^i$$

$$n \le 10^6, q \le 10^5$$

棋盘

NiroBC 姐姐有一个漂亮的大棋盘，棋盘的长和宽分别是 N 和 M ，有 $N \times M$ 个格子。NiroBC 姐姐有无限数量的黑，白两种颜色的棋子，她关心的是，如果每个格子都放上恰好一个棋子，那么所有可能的局面当中，所有黑子构成的联通块数量正好为 K 的局面有多少种。

两个局面被视为不同，当且仅当存在一个位置，在这两个局面中放了不同的棋子。

两个格子被视为相连，当且仅当它们有一条公共边，且它们的棋子同色。

对于所有数据，N , M 为正整数，$1 \le N \le 3$ ，$0 \le K \le N \times M$ 。

当 $N = 1$ 时，$1 \le M \le 10^5$ 。

当 $N = 2$ 时，$1 \le M \le 5 \times 10^4$ 。

当 $N = 3$ 时，$1 \le M \le 10^4$ 。

任意基DFT

给定 n 次多项式

$$f(x) = \sum_{i=0}^{n} a_i x^i$$

Q 次询问，第 i 次询问 $f(q_i)$ 对 998244353 取模的值。

输入格式

由于本题的数据范围较大，所有测试点的 q_i 将在程序内生成。

第一行两个整数 n, Q。n, Q 的意义见题目描述。

第二行 $n+1$ 个整数，第 i 个整数表示 a_{i-1}

第三行三个整数 $q_0, q_{\mathrm{x}}, q_{\mathrm{y}}$，表示 q_i 的生成方式。

q_i 按照如下规则生成:

$\forall 1 \le i \le Q, q_i = (q_{i-1} \times q_{\mathrm{x}} + q_{\mathrm{y}}) \bmod 998244353$

[THUWC2018] D2T2

- 给n种物品，第i种有a_i个。问任意排列物品，不存在任和一个非空前后缀中每种个数相同的方案数。$n \leq 100, \max(a_i) \leq 2 * 10^5$

斯特林容斥

- 列斯特林数的EGF:

$$\sum_{i\ge k}\begin{bmatrix} i \\ k \end{bmatrix}\frac{z^i}{i!}=\frac{(\ln\frac{1}{1-z})^k}{k!}$$
$$\sum_{i\ge k}\begin{Bmatrix} i \\ k \end{Bmatrix}\frac{z^i}{i!}=\frac{(e^z-1)^k}{k!}$$

- 对于一个有序集合，我们可以任意划分，考虑两类计数问题:
- 1.一个划分的代价只跟划分出的集合个数有关，且对于一个“划分方案”，我们可以确定不在同一集合中的元素确实不在同一集合。
- 典型例子：要求图连通
- 2.一个划分的代价只跟划分出的每个集合大小有关（是不同集合大小的对应贡献的乘积），且对于一个“划分方案”，我们可以确定在同一集合中的元素确实在同一集合。
- 典型例子：要求元素互不相同

斯特林容斥

- 1.我们枚举一种“划分方案”，使得里面每个集合是一个真实方案中若干集合的并。则贡献只跟这个“划分方案”的集合个数有关。
- 具体计算的时候，考虑原先关于集合个数的贡献的EGF是$F(z)$，新的“划分方案”对应的EGF是$G(z)$，有 $F(z)=G(e^z-1)$，则
- $G(z)=F(\ln(1+z))$。当问题是图连通的时候，有$F(z)=z$，此时个数为i的贡献为 $(-1)^{i-1}(i-1)!$。

斯特林容斥

- 2.我们枚举一种“划分方案”，使得真实方案中每个集合是里面若干集合的并。则贡献只跟这个“划分方案”中每个集合大小对应贡献的乘积有关。
- 具体计算的时候，考虑原先关于集合大小的贡献的EGF是$F(z)$，新的划分方案对应的EGF是$G(z)$，同样有$F(z)=G(e^z-1)$，即
- $G(z)=F(\ln(1+z))$ 。当问题是元素互不相同的时候,$F(z)=z$，此时集合大小为i的贡献是$(-1)^{i-1}(i-1)!$。

U群把妹王

有 $n \times m$ 个格子，每个格进行染色，可以选择 k 种颜色之一。对于集合 S, T，你需要计数有多少种格子的染色方案，满足：

- 对于每一行的图案拿出来，和它相同的图案总共有 r 行（含自身），则 $r \in S$。
- 对于每一列的图案拿出来，和它相同的图案总共有 c 列（含自身），则 $c \in T$。

答案对 $P = 998244353$ 取模。

为了让这道题看起来代码比较健康，保证 $1 \in S \cap T$。

对于 100% 的数据，保证 $1 \le n, m \le 10^5, 1 \le k \le P-1, 1 \le a, b \le 5, 1 \in S \cap T, S \subseteq [1, n], T \subseteq [1, m]$。

一个ODE的题

Description

有一个随机变量 z, 初始 $z = 0$.

执行 n 次操作: 每次操作是从 0 到 k 中以 p_t 的概率选择一个整数 t, 并令 $z := z + t$.

求 $\min(z, x)$ 的期望. 答案模 998244353.

对于 100% 的数据, 满足 $1 \le n \le 10^7, 1 \le k \le 100, 1 \le x \le \min(10^7, \frac{5\times 10^7}{k}), 1 \le a_i \le 100$.

Simple permutation

- 求长度为n的不存在非平凡子序列的排列数目。
- 扩展：快速求1~n的答案。

Simple permutation

特判$n = 1, 2$。

掌握了析合树的相关知识后，我们发现当$n > 3$时一个“正确”排列，等价于一棵根节点为析点，且有n个叶节点作为儿子的析合树。那么我们的问题就可以从排列数目计数变为对析合树的计数了。

为方便描述，与课件中不同，析合树上的叶节点我们认为它既不是析点也不是合点。

我们可以发现析合树有这样的性质：如果一个点是某个析点的儿子，它自己的属性没有任何限制，即它自己可以是任意的析点、合点或叶节点；如果一个点是某个合点的儿子，而它父亲合点的儿子值单调上升/下降，那么它自己仍然可以是析点或叶节点，但是如果是合点，自己的儿子值只能单调下降/上升。

设$F(x)$表示$n > 3$的“正确”排列数目对应的生成函数；$G(x)$表示根节点为合点，且根节点儿子值单调上升/下降的排列数目对应的生成函数，显然上升/下降的生成函数是相等的；$H(x)$表示所有排列数目对应的生成函数，即$H(x) = \sum_{i \geq 1} i! x^i$。显然，答案就是$[x^n]F(x)$。

那么有$G(x) = \sum_{i \geq 2} (H(x) - G(x))^i$，是因为如果根节点是儿子值单调上升/下降的合点，那么不能存在某个根节点的儿子是儿子值单调上升/下降的合点，因此每个儿子的生成函数就是$H(x) - G(x)$，而根节点的儿子数目是≥ 2的任意正整数。

即$G(x) = \frac{H^2(x)}{H(x)+1}$，这个可以多项式求逆来快速求出。

Simple permutation

我们可以发现，根节点为析点的排列数目对应的生成函数为$F(H(x))$（这里的意思是$F(x)$与$H(x)$的函数复合）。具体怎么理解呢？可以考虑$F(x)$对应的析合树，我们可以把根节点（析点）的每个叶节点儿子替换成任意的一棵析合树，来得到所有根节点为析点的排列对应的生成树，即将$H(x)$代入$F(x)$。于是有$F(H(x))+2G(x)+x=H(x)$，这里的意思是一个排列对应的析合树要么是一个孤立的根节点（x），要么根节点是析点（$F(H(x))$），要么根节点是儿子值单调上升/下降的合点（都是$G(x)$）。

转化一下有$F(H(x))=H(x)-x-2G(x)$。设$P(x)=H(x)-x-2G(x)$，那么变为$F(H(x))=P(x)$，那么如何得到$[x^n]F(x)$呢？

这是个经典问题，设$H^{-1}(x)$表示$H(x)$的复合逆，即$H(H^{-1}(x))=1$，那么将$H^{-1}(x)$代入，有$F(x)=P(H^{-1}(x))$。

考虑扩展拉格朗日反演，$[x^n]P(H^{-1}(x))=\frac{1}{n}[x^{-1}](P'(x)*\frac{1}{H^n(x)})$，给多项式求导是简单的，然后直接多项式快速幂即可。

时间复杂度$O(nlogn)$。

地底蔷薇

给定集合S，请你求出n个点的“所有极大点双连通分量的大小都在S内”的不同简单无向连通图的个数对998244353取模的结果。

点双连通分量：删去任意一个点后剩下的点依然保持连通的连通子图。
极大点双连通分量：一个点双连通分量，且不存在更大的点双连通分量包含自己。
极大点双连通分量的大小：指连通分量包含的点数。
两个简单无向图不同，当且仅当存在某条边(u,v)出现在了其中一个无向图，而没有出现在另一个无向图。

对于10%的数据，$n \le 6$
对于30%的数据，$n \le 12$
对于50%的数据，$n \le 1000$
对于100%的数据，$n \le 10^5, (\sum_{x \in S} x) \le 10^5$

无标号树计数

- 计数n个点有根/无根的无标号树个数（两棵树不同当且仅当不同构）。

牛顿恒等式

- 对于多项式$\sum_{i=0}^{n} a_{n-i} z^i$的n个根$x_1, x_2, ..., x_n$，定义幂和 $S_k = \sum_{i=1}^{n} x_i^k$
- 有牛顿恒等式

$$\begin{cases} \sum_{i=0}^{n} a_i S_{d-i} = 0 \quad (d > n) \\ (\sum_{i=0}^{d-1} a_i S_{d-i}) + d a_d = 0 \quad (1 \le d \le n) \end{cases}$$

牛顿恒等式

- 证明考虑构造生成函数 $F(z)=\sum_{i=1}^{n}\frac{x_i}{1-x_iz}=\sum_{i\geq 0}S_{i+1}z^i$
- 以及 $G(z)=\sum_{i=0}^{n}a_iz^i=\prod_{i=1}^{n}(1-x_iz)$
- 那么显然有 $F(z)=\frac{-G'(z)}{G(z)}$，也即 $F(z)G(z)+G'(z)=0$。
- 逐项对比系数即得证。
- 这个定理联系了根的幂和与多项式系数，得到其中任一个都可以在$\mathcal{O}(n\log n)$ 时间内得到另一个。
- 有些问题上会有应用。

多项式求和

给定 n, m, a, b, c
定义 $f_0(x) = 1$
对于 $k > 0$, $f_k(x) = \sum_{i=0}^{x}(ai^2 + bi + c)f_{k-1}(i)$
对于 $\forall 0 \leq i \leq n$, 求 $f_i(m) \bmod 1004535809$

$0 \leq n \leq 250000$

$0 \leq m \leq 10^9$

$1 \leq a, b, c \leq 10^9$

小Q的序列

小 Q 喜欢在序列上数数。

定义一个序列 $a_{1...k}$ 的权值为 $\prod_{i=1}^{k}(a_i+i)$。

现在小 Q 有一个长度为 n 的序列 $c_{1...n}$。他想知道他的序列的所有 2^n-1 个非空子序列的权值和。
由于答案很大，你只需要输出答案对 998244353 取模的结果。

新年的追逐战

但两只鞋太太的家实在太大了，为了简单地说明，定义两个简单无向图 $G_1=(V_1,E_1),G_2=(V_2,E_2)$ 的乘积为一个新的图 $G_1\times G_2=(V^*,E^*)$。

其中新的点集 V^* 为:

$$V^*=\{(a,b)|a\in V_1,b\in V_2\}$$

其中新的边集 E^* 为:

$$E^*=\{((u_1,v_1),(u_2,v_2))\mid (u_1,u_2)\in E_1,(v_1,v_2)\in E_2\}$$

对于正整数 n，以及给定的图 $G_1,G_2,\dots,G_n$，两只鞋太太的家可以表示成

$$H=(((G_1\times G_2)\times G_3)\times\cdots)\times G_n$$

为了方便逃亡（戏弄汤姆），对于同一个连通块，杰瑞事先打通了所有的老鼠洞，你只需要计算一遍。

为了方便逃亡（戏弄汤姆），对于同一个任何一个连通块，都有老鼠洞。

也就是说，你需要求的是 H 的连通块数量。但是杰瑞忘记了每个 G_k 的具体细节，所以我们现在假设每个 G_k 中任意两点间都有 $\frac{1}{2}$ 的概率连边，求 H 的连通块的期望。显然 G_k 的全体取法共有 $2^{\binom{m_1}{2}+\binom{m_2}{2}+\cdots+\binom{m_n}{2}}$ 种。

方便起见，你只需要输出答案乘以 $2^{\binom{m_1}{2}+\binom{m_2}{2}+\cdots+\binom{m_n}{2}}$，对 998244353 取模即可。

Thanks