Maths

yanQval

Tsinghua, IIIS

2020 年 7 月 23 日

埃氏筛法

先将 1 筛去，然后每次选择一个未被筛去且未被标记过的数 x，将 x 标记为质数，同时筛去 x 的倍数。
复杂度为 $O(n \log \log n)$。

欧拉筛法

建立一个 prime list，先将 1 筛去，从小到大枚举每一个数 x 。
如果 x 没有被筛去，那么将 x 加入 prime list。
然后枚举 prime list 里的质数 p ，将 xp 筛去，当 $p \mid x$ 时结束。
复杂度为 $O(n)$ ，因为每个合数只会在当 p 为其最小质因子时筛去。

质数分布

平均每 $O(\log n)$ 个数中，就有一个质数。
$\leq n$ 的数中，质数的个数是 $O(\frac{n}{\log n})$ 。

质数判断

朴素的是 $O(\sqrt{n})$ 。更高效的算法是 Miller-Robin 算法。

质数判断

朴素的是 $O(\sqrt{n})$ 。更高效的算法是 Miller-Robin 算法。
若 p 是质数，对于 $x \in (0, p)$ ，方程 $x^2 \equiv 1 \pmod p$ 的解为 $x = 1$ *or* $p - 1$ 。
令 $p - 1 = m * 2^q$ ， m 为奇数。
如果 $a^m, a^{m*2}, \dots, a^{m*2^q}$ 中没有 1 或者存在 1 但前一个数不为 1 或 $p - 1$ ，则 p 不为质数。
在 p 不为质数的情况下，随机选择一个 a ，错误概率 $< \frac{1}{2}$ ，所以重复尝试 $10 \sim 20$ 次即可。也可以是直接选取前 $10 \sim 20$ 个质数。

辗转相除法

$$gcd(x, y) = \begin{cases} x & y == 0 \\ gcd(y, x \bmod y) & y \neq 0 \end{cases}$$

EXGCD

求解 $ax + by = gcd(a, b)$ 。
假设已知 $bx' + (a \bmod b)y' = gcd(a, b)$ 。
因为 $a \bmod b = a - \lfloor \frac{a}{b} \rfloor b$ 。
所以 $bx' + (a - \lfloor \frac{a}{b} \rfloor b)y' = gcd(a, b) \Rightarrow ay' + b(x' - \lfloor \frac{a}{b} \rfloor y') = gcd(a, b)$ 。
即 $x = y', y = x' - \lfloor \frac{a}{b} \rfloor y'$ 。
时间复杂度 $O(\log n)$ 。

快速幂

$$a^b = \begin{cases} (a^2)^{\frac{b}{2}} & b \text{ is even} \\ a * (a^2)^{\frac{b-1}{2}} & b \text{ is odd} \end{cases}$$

逆元

x 在模 m 意义下的逆元 y 满足 $xy \equiv 1 \pmod m$ 。
存在逆元的条件是 $(x, m) = 1$ 。
求逆元的方法有三种：利用 EXGCD，利用费马小定理，线性预处理。

EXGCD 求逆元

$$xy \equiv 1 \pmod m$$

$$xy + am = 1$$

解出这个方程的一组解 (y, a) 即可。

费马小定理求逆元

费马小定理：当 m 为质数时，$a^{m-1} \equiv 1 \pmod m$ 。
所以 x 的逆元就为 x^{m-2} 。

线性预处理逆元

下面均是在模 p 意义下进行运算。

$$i * \lfloor \frac{p}{i} \rfloor + (p \bmod i) \equiv 0$$

$$-i * \lfloor \frac{p}{i} \rfloor \equiv (p \bmod i)$$

$$i * (-\lfloor \frac{p}{i} \rfloor * (p \bmod i)^{-1}) \equiv 1$$

所以 $i^{-1} = -\lfloor \frac{p}{i} \rfloor * (p \bmod i)^{-1}$ 。

CRT

假设 $m_1, m_2, \ldots, m_k$ 两两互质，并记 $M = \prod_{i=1}^{k} m_i$ ，则同余方程组：

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \qquad \ldots \\ x \equiv a_k \pmod{m_k} \end{cases}$$

在模 M 意义下有唯一解。
这个解 $x \equiv \sum_{i=1}^{k} t_i a_i M_i \pmod{M}$ ，其中 $M_i = \frac{M}{m_i}, t_i \equiv M_i^{-1} \pmod{m_i}$ 。

欧拉函数

定义 $\phi(n)$ 为 $\leq n$ 的数中与 n 互质的数的个数，即 $\phi(n) = \sum_{i=1}^{n}[(i,n)=1]$ 。
计算式：$\phi(x) = x(1-\frac{1}{p_1})(1-\frac{1}{p_2})\dots(1-\frac{1}{p_k})$ ，其中 $p_1, p_2, \dots, p_k$ 是 x 的不同质因子。
递推式：$\phi(pq) = \phi(p) * \phi(q)$ 当 p, q 互质时。利用这个可以在欧拉筛法的同时预处理出 $\leq n$ 的所有数的 ϕ 。

欧拉定理

若 $(a,p)=1$ 则 $a^{\phi(p)}\equiv 1 \pmod p$ 。
如果 $(a,p)\neq 1$ 呢?

$$a^b=\begin{cases} a^b & (b\le\phi(n)) \\ a^{(b \bmod \phi(n))+\phi(n)} & (b>\phi(n)) \end{cases} \pmod n$$

莫比乌斯反演

若 $F(n) = \sum_{d|n} f(d)$ ，则有 $f(n) = \sum_{d|n} \mu(d) F(\frac{N}{d})$ 。

$$\mu(d) = \begin{cases} 1 & n = 1 \\ (-1)^k & n = p_1 p_2 \dots p_k, \forall i \neq j, p_i \neq p_j \\ 0 & others \end{cases}$$

欧拉函数、莫比乌斯函数性质

$$\sum_{d|n} \phi(d) = n$$

$$\sum_{d|n} \mu(d) = [n == 1]$$

$$\sum_{d|n} \frac{\mu(d)}{d} = \frac{\phi(n)}{n}$$

BSGS

求解 $a^x \equiv b \pmod c$ 。
令 $x = p * g - m$ ，那么

$$a^{p*g-m} \equiv b \pmod c$$

$$(a^g)^p \equiv b * a^m \pmod c$$

那么我们只需要把右边 $0 \le m < g$ 的值预处理出来，然后再枚举 p 的值并查表即可。复杂度是 $O(g + \frac{c}{g})$ ，g 取 $O(\sqrt{c})$ 时复杂度最低，为 $O(\sqrt{c})$ 。

排列

从 n 个数中有序地选出 m 个数的方案数，用 $P(n, m)$ 表示。

$$P(n, m) = n * (n-1) * (n-2) * \cdots * (n-m+1) = \frac{n!}{(n-m)!}$$

组合

从 n 个数中无序地选出 m 个数的方案数，用 $C(n,m)$ 表示。

$$C(n,m)=\frac{A(n,m)}{m!}=\frac{n!}{m!(n-m)!}$$

二项式定理

$$(a+b)^n = \sum_{i=0}^{n} C(n,i) a^i b^{n-i}$$

例题 1

将 n 个不同的球放到 m 个不同的袋子里有多少种方案?

例题 1

将 n 个不同的球放到 m 个不同的袋子里有多少种方案?
直接就是 m^n 。

例题 2

将 n 个相同的球放到 m 个不同的袋子里有多少种方案?

例题 2

将 n 个相同的球放到 m 个不同的袋子里有多少种方案?
考虑从一排 $n+m-1$ 个元素中选出 $m-1$ 个，这 $m-1$ 个元素将序列分成了 m 段，第 i 段的元素个数就是第 i 个袋子中球的个数。
不难发现选元素的方案和放球的方案一一对应，因此方案数就是 $C(n+m-1,m-1)$。

例题 3

将 n 个相同的球放到 m 个相同的袋子里有多少种方案?

例题 3

将 n 个相同的球放到 m 个相同的袋子里有多少种方案?
由于袋子是相同的，我们通过保证球数是单调不减的来防止重复统计。
用 $f[i][j]$ 表示将 i 个相同的球放到 j 个相同的袋子里的方案数。
考虑第 1 个袋子是否放球，如果放的话，由于球数单调不减，我们必须在每个袋子里都放一个球。如果不放的话，那我们直接考虑后面的袋子。
$f[i][j] = f[i-j][j] + f[i][j-1]$。
时间复杂度 $O(nm)$ 。

Lucas 定理

$$\binom{n}{m} = \binom{\lfloor \frac{n}{p} \rfloor}{\lfloor \frac{m}{p} \rfloor} \binom{n \bmod p}{m \bmod p} \pmod p$$

其中 p 是质数。

抽屉原理

把 $n+1$ 个东西放入 n 个抽屉，必有一个拥有至少两个东西。
扩展：把 m 个东西放入 n 个抽屉，至少有一个抽屉拥有至少 $\lceil \frac{m}{n} \rceil$ 个东西，至少有一个抽屉拥有至多 $\lfloor \frac{m}{n} \rfloor$。

加法原理

完成 A 有 n 种方法，完成 B 有 m 种方法，那么完成 AB 中的一个有 $n+m$ 种方法。

乘法原理

完成 A 有 n 种方法，完成 B 有 m 种方法，那么先完成 A 再完成 B 有 nm 种方法。

容斥原理

记 $f(S)$ 为满足集合 S 中至少一个条件的方案数，$g(S)$ 表示满足集合 S 中所有条件的方案数。
那么有 $f(S) = \sum_{T \subseteq S, |T| \neq 0} (-1)^{|T|-1} g(T)$ 。
记 $h(S)$ 表示不满足集合 S 中所有条件的方案数。
那么有 $g(S) = \sum_{T \subseteq S} (-1)^{|T|} h(T)$ 。

概率

概率空间

样本空间 Ω，事件集合 F，概率测度 P。

概率

概率空间

样本空间 Ω，事件集合 F，概率测度 P。

事件是 Ω 的一个子集。所有事件的集合为 F。

概率

概率空间

样本空间 Ω，事件集合 F，概率测度 P。

事件是 Ω 的一个子集。所有事件的集合为 F。
概率测度 P 是 F 到 $\mathbb{R}$ 的一个函数。合理的概率测度，需要满足以下 3 条概率公理：
(1) 对于任意的事件 A，有 $P(A) \geq 0$（非负性）；
(2) $P(\Omega) = 1$（规范性）；
(3) 对于事件 A 和 B，如果 $A \cap B = \Phi$，有 $P(A \cap B) = P(A) + P(B)$（可加性）。

概率

条件概率

$$P(A \mid B) = \frac{P(AB)}{P(B)}$$

概率

条件概率

$$P(A \mid B) = \frac{P(AB)}{P(B)}$$

全概率公式

如果 $B_1, B_2, B_3, \ldots, B_n$，那么有：

$$P(A) = \sum_k P(A \mid B_k) P(B_k)$$

概率

Bayes 公式

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

概率

Bayes 公式

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

$$P(A_k \mid B) = \frac{P(B \mid A_k)P(A_k)}{\sum_j P(B \mid A_j)P(A_j)}$$

随机变量与期望

随机变量的定义

函数 $X : \Omega \to \mathbb{R}$ 被称为一个随机变量。

随机变量与期望

随机变量的定义

函数 $X:\Omega\to\mathbb{R}$ 被称为一个随机变量。

随机变量的期望

$$E[X]=\sum_{\omega}P(\omega)X(\omega)=\sum_{x}xP(X=x)$$

这里 $X=x$ 表示的是一个事件，等价于 $\omega\mid\omega\in\Omega,X(\omega)=x$。

随机变量与期望

随机变量的独立性与乘积的期望

对于两个随机事件 X_1, X_2 和实数 $x1 \in X1(\Omega), x2 \in X_2(\Omega)$，如果有 $P(X_1 = x_1, X_2 = x_2) = P(X_1 = x_1)P(X_2 = x_2)$ 就称 X_1, X_2 相互独立。
两个独立的随机变量的积的期望等于期望的积。

$$E[X_1X_2] = E[X_1]E[X_2]$$

期望的线性性

$$E[\alpha X_1 + \beta X_2] = \alpha E[X_1] + \beta E[X_2]$$

全期望公式

$$P((X \mid A) = x) = \frac{P(X = x, A)}{P(A)}$$

全期望公式

$$P((X \mid A) = x) = \frac{P(X = x, A)}{P(A)}$$

$$E[E[X \mid Y]] = E[X]$$

概率转移网络

状态转移网络是一个有向网络，由点集（状态集）V，转移概率矩阵 $G: V \times V \to [0,1]$，以及起点 v_0 组成。
其中，$\forall u \in V, \sum_v G[u,v] \le 1$。

CF 113D

一张图，A，B 两个人初始在 a,b 两点。每分钟每个人有 p_i 的概率不动，有 $1 - p_i$ 的概率等概率随机移动到一个相邻点，i 是当前点的编号。两个人只能在点处相遇，不能再边上相遇。问在每个点相遇的概率。

CF 113D

一张图，A，B 两个人初始在 a,b 两点。每分钟每个人有 p_i 的概率不动，有 $1-p_i$ 的概率等概率随机移动到一个相邻点，i 是当前点的编号。
两个人只能在点处相遇，不能再边上相遇。问在每个点相遇的概率。
状态集 V 为 $V_0 \times V_0$，V_0 为题目中点的集合。
转移概率矩阵不难得到。$v_0 = (a, b)$。
再定义停止状态集合 $S = (a, a) \mid a \in V_0$。

解法一: 迭代法

将 S 中状态的转移概率特殊处理，将其转移概率除了转移到自己为 1 以外其它的全部为 0。
记网络中时刻 t 时，质点位于每个点的概率为 x_i^t，不难发现 $x_{t+1} = x^t G$。
而我们需要的是 x^∞，当 t 足够大时，$x^t \approx x^\infty$。这个是可以用快速幂来尽可能高地算出 G^{2^K}。

解法二：解线性方程组

将 S 中状态的转移概率全部设为 0。
设质点停留在每个点的次数的期望 E_u。

$$E_u = x_u^0 + \sum_{t=1}^{\infty} x_u^t = x_u^0 + \sum_{t=1}^{\infty} \sum_{v} x_v^{t-1} G[v, u]$$

$$= x_u^0 + \sum_{v} G[v, u] \sum_{t=0}^{\infty} x_v^t = x_u + \sum_{v} G[v, u] E_v$$

两个常用不等式

Markov 不等式

$$P(X \geq a) \leq \frac{E[x]}{a}$$

大多数情况下取不到等号，并且左边会远小于右边。
应用：对算法运行时间的估计。

Chebyshev 不等式

称随机变量 X 的方差 $Var[x]$ 为 $E[(x - E[x])^2]$，标准差 σ 为 $\sqrt{Var[X]}$。

$$P(|X - E[x]| \geq a) \leq \frac{Var[x]}{a^2}$$

$$P(|X - E[x]| \geq c\sigma) \leq \frac{1}{c^2}$$

错位排列

有多少个长度为 n 的排列 P 满足对于任意的 i 有 $p_i \neq i$。
$n \leq 10^5$

用 $f[i]$ 表示 i 个数的错位排列个数。
假设 $P_j = i$ 。如果 $P_i = j$ ，那么剩下的 $i-2$ 个数构成错位排列。否则交换 p_i 和 p_j ，前面 $i-1$ 个数构成错位排列。

$$f[i] = f[i-2]*(i-1) + f[i-1]*(i-1)$$

时间复杂度 $O(n)$。

考虑容斥原理，枚举哪些位置满足 $p_i = i$ 。
设有 i 个位置满足，那么选出 i 个位置的方案是 $C(n,i) = \frac{n!}{i!(n-i)!}$，这 i 个位置满足条件的方案数是 $(n-i)!$，乘起来就是 $n!/i!$ 。
那么答案就是 $\sum_{i=0}^{n}(-1)^i\frac{n!}{i!}$ 。
时间复杂度 $O(n)$ 。

硬币购物

有 k 种硬币，面值分别为 $c_1, c_2, \ldots, c_k$ 。某人去商店买东西，去了 m 次，每次带 d_i 枚 c_i 硬币，买价值为 n 的东西。请问每次有多少种付款方法。
$n \leq 10^5, m \leq 1000, k \leq 4$

令 $f[i]$ 表示不考虑 d 的限制下，购买价值为 i 的东西的方案数。

$$f[i] = f[i - c_1] + f[i - c_2] + \cdots + f[i - c_k]$$

根据容斥原理，我们可以将“所有硬币不超过限制”转化为枚举“一些硬币超过限制”。

超过限制很好处理，只要将这种硬币强制使用 d_i 个，然后就变成没有限制了。

时间复杂度 $O(nk + m * 2^k)$ 。

RedIsGood

桌面上有 n 张红牌和 m 张黑牌，随机打乱顺序后放在桌面上，开始一张一张地翻牌，翻到红牌得到 1 美元，黑牌则付出 1 美元。可以随时停止翻牌，在最优策略下期望能得到多少钱。

$n, m \leq 1000$

用 $f[i][j]$ 表示剩下 i 张红牌和 j 张黑牌获得钱的期望。

$$f[i][j] = max(0, (f[i-1][j]+1) * \frac{i}{i+j} + (f[i][j-1]-1) * \frac{j}{i+j})$$

时间复杂度 $O(nm)$ 。

GCD SUM

求 $\sum_{i=1}^{n} \sum_{j=1}^{m} gcd(i, j)$ 。
$n, m \leq 10^6$

令 $f[i]$ 为 gcd 是 i 的倍数的数对的个数，那么 $f[i] = \lfloor \frac{n}{i} \rfloor * \lfloor \frac{m}{i} \rfloor$。
令 $g[i]$ 为 gcd 等于 i 的数对个数，因为 $f[i] = g[i] + g[2i] + g[3i] + \ldots$，
所以 $g[i] = f[i] - g[2i] - g[3i] - \ldots$。
有了 g 之后就可以直接计算答案了。

经典题

求 $\sum_{i=1}^{n}\lfloor\frac{n}{i}\rfloor$ 和 $\sum_{i=1}^{n}(n \bmod i)$ 。
$n \leq 10^9$

当 $i \leq \sqrt{n}$ 时暴力；当 $i > \sqrt{n}$ 时 $\frac{n}{i} \leq \sqrt{n}$，枚举 $\lfloor \frac{n}{i} \rfloor$ 的值即可。
$n \bmod i = n - \lfloor \frac{n}{i} \rfloor * i$，类似分段计算即可。

Endless Punishment

S_1, S_2 是 $\{1, 2, 3, \ldots, n\}$ 的两个子集。
有一个长度为 n 的 01 序列，每次按以下规则生成下一个序列：

- 查看下标在 S_1 中的元素有多少个 0 ，如果是奇数个则在序列后添加一个 1 ，否则添加一个 0 ；
- 删除序列的第一个数；
- 将下标在 S_2 中的元素取反；
- 例如：$S_1 = \{1, 3\}, S_2 = \{2, 4\}$ ，原串为 1000 ，则变化过程为：$1000 \to 10001 \to 0001 \to 0100$ 。

给出 S_1, S_2 ，初始状态 s 和目标状态 t ，问最少多少步能从初始状态变成目标状态。
$n \leq 31$

显然答案不会超过 2^n 。
可以构建一个 $(n+1)*(n+1)$ 的矩阵 M , 使得对于任一序列 s 的下一个序列为 sM 。
那么要求的就是 $sM^x=t$ 的最小解。
BSGS。复杂度 $O(n^2 2^{\frac{n}{2}})$ 。

Fast wyh2000 Transform

给出两个模 3 意义下的下标为 $0 \sim n-1$ 的数列 A, B，求：
$c_i \equiv \sum_{j+k=i} \binom{i}{j} a_j b_k \pmod 3$
$n \leq 50000$

满足 $\binom{i}{j} \neq 0 \pmod 3$ 的 (i, j) 数量，因为转成三进制后 i 的每一位都要 $\geq j$ 的对应位。
直接按位枚举即可。
复杂度 $O(n^{\frac{\ln 6}{\ln 3}})$ 。

Privateparty

N 个参加聚会，和一个数组 a，a_i 表示第 i 个人讨厌的人，如果一个到聚会门口的时候发现他讨厌的人已经在聚会里面，则他不会参加聚会，否则他会参加聚会。$a_i = i$ 表示他没有讨厌的人。N 个人来的先后顺序是任意的，也就是说 n 个来的先后顺序构成的 1 到 n 的排列是任意的。问参加聚会的人的期望是多少?

一个人是否参加舞会，会受到一些人的影响，这些人是：$w_1, w_2, w_3, \ldots, w_k$ ，其中 $w_1 = i$ ，w_1 讨厌 w_2 ，w_2 讨厌 w_3 ，……，w_{k-1} 讨厌 w_k，w_k 讨厌 w_1 到 w_k 的某个人。
那么他参加的概率就是：$1 - ($$w_2$ 比 w_1 先来的概率 $) + ($$w_3$ 比 w_2 先来且 w_2 比 w_1 先来的概率 $) - \ldots$。

无向联通图计数

n 个点的带标号无向联通图计数。
$n \leq 1000$

经典题

从 $(0,0)$ 走到 (n,m) ，每次只能 $x++$ 或 $y++$ 。
有 k 个不可经过点 (x_i,y_i) 。
求方案数。
$n,m \le 10^5, k \le 1000$

方程

求方程 $x_1 + x_2 + \cdots + x_n = m$ 的非负整数解数。
要求对于 $1 \le i \le n_1, x_i \le A_i$ 。
对于 $n_1 < i \le n_1 + n_2, x_i \ge B_i$ 。
$n, m \le 10^6, n_1, n_2 \le 8$

某 noip 模拟题

$n * m$ 的棋盘上放棋子，棋子有 c 种颜色，每个格子最多放一个棋子。要求每行每列至少一个，整个棋盘上每种颜色至少一个。求方案数。
$n, m, c \leq 1000$

and

将 $\{a_1, a_2, \ldots, a_n\}$ 分成两个集合 S_1, S_2 ，使得两个集合元素的 and 和相同。求方案数。
$n \leq 50, a_i < 2^{20}$

substring pairs 简化

给出一个字符串 T，求有多少个长度为 n 的串 S 满足 S 包含 T。
$n \leq 100$

substring pairs 简化

给出一个字符串 T，求有多少个长度为 n 的串 S 满足 S 包含 T。
$n \le 100$
$n \le 1000$

YetAnotherNim

现在有一个博弈游戏。
有 n 堆石子，每堆石子的数量在 $1 \sim m$ 之间，其中 $m+1=2^u$。
先手先从中选出连续 K 堆石子，删掉其他的所有堆。
后手接着删去任意堆石子，可以不删，但是不能全删。
然后两人开始玩 NIM 游戏。
求后手必胜的初始局面数量。
$n, m \leq 10^9$

题目等价于求长度为 n，每个数均在 $1 \sim m$ 之间，任意连续 K 个数线性相关的序列个数。
显然当 $K > log_2(m+1)$ 时，任意 K 个数肯定线性相关，答案就是 m^n。
当 $K <= log_2(m+1)$ 时，补集转化为求存在某 K 个数线性无关的序列个数。
设 $f_{i,j}$ 表示考虑了前 i 个数，且恰好后 j 个数线性无关的序列个数。

$$f_{i+1,j+1} + = f_{i,j} * (m + 1 - 2^j)$$

$$f_{i+1,j+1} + = f_{i,k} * 2^j$$

$$Ans + = f_{i,k} * m^{n-i}$$

矩阵乘法优化即可。

TrichyInequality

求出满足 $\sum_{i=1}^{m} x_i \le s, \forall i \le m, x_i > 0, \forall i \le n, x_i \le t.$ 的向量 X 的解数。

TrichyInequality

求出满足 $\sum_{i=1}^{m} x_i \le s, \forall i \le m, x_i > 0, \forall i \le n, x_i \le t.$ 的向量 X 的解数。
$m \le 10^9, max(1, m-100) \le n \le m, t \le 10^5, s \le 10^{18}$

首先我们枚举前 n 个 x 的取值。
则根据简单的组合数学原理我们知道答案是 $\sum_{\sum_{i=1}^{n} x_i=k,\forall i,1\le x_i\le t}\binom{s-k}{m-n}$.
注意到那个组合数对于 k 是一个 $m-n$ 次多项式，我们可以比较方便地展开并求出每一项的系数。
将系数提前，就变成了 $\sum_{i=1}^{m-n} a_i * \sum_{\sum_{i=1}^{n} x_i=k,\forall i,1\le x_i\le t} k^i$.
如何求 k^i?
不妨设 $f_{n,m}=\sum_{\sum_{i=1}^{n} x_i=k,\forall i,1\le x_i\le t} k^m$.
可以很容易地将 k 写成 $k'+x_n$，然后二项式展开得到递推式。
矩乘加速将 $f_{n,0}\sim f_{n,m-n}$ 求出来即可。