容斥与反演

Lime

YALI 2019.7

目录

一 容斥原理

容斥原理是一种重要的计数公式, 定义如下:

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{k=1}^{n}(-1)^{k-1}\sum_{1\le i_1<i_2<\cdots<i_k\le n}\left|A_{i_1}\cap A_{i_2}\cap\cdots\cap A_{i_k}\right|$$

一 容斥原理的简单证明

将问题转为求 有n场比赛, 每个人拿过一些比赛的金牌, 试统计拿过金牌的人数.

设 A_i 表示拿过第i场比赛金牌的人. 现在考虑一个人产生的贡献.

若一个人共获得了k块金牌, 根据公式, 他的贡献显然有:

$$\sum_{i=1}^{k}(-1)^{i+1}\binom{k}{i}$$

显然, 当k=0时它的贡献为0. 我们要证明k≠0时, 它的贡献为1.

— 容斥原理的简单证明

$$
\begin{aligned}
&\sum_{i=1}^{k}(-1)^{i+1}\binom{k}{i}\\
=&\sum_{i=1}^{k}(-1)^{i+1}(\binom{k-1}{i-1}+\binom{k-1}{i})\\
=&\sum_{i=1}^{k}(-1)^{i+1}\binom{k-1}{i}+\sum_{i=1}^{k}(-1)^{i+1}\binom{k-1}{i-1}\\
=&\sum_{i=1}^{k}(-1)^{i+1}\binom{k-1}{i}+\sum_{i=1}^{k-1}(-1)^{i+2}\binom{k-1}{i}+\binom{k-1}{0}\\
=&1
\end{aligned}
$$

一 容斥与dp

求有多少排列满足 $a_i \neq i$

我们可以看做有n个条件限制, 第i个条件是 $a_i \neq i$

不等于是个很大的条件，除了i都能取，而等于是个很严格的条件，直接帮助我们确定了一些位置.

于是设dp[i]表示n=i时的答案, 那么有:

$$dp[n] = \sum_{i=0}^{n} (-1)^i \cdot \binom{n}{i} \cdot (n-i)!$$

$$dp[n] = \sum_{i=0}^{n} (-1)^i \cdot \frac{n!}{i!}$$

$$= (-1)^n + \sum_{i=0}^{n-1} (-1)^i \cdot \frac{n!}{i!}$$

$$= (-1)^n + n \cdot \sum_{i=0}^{n-1} (-1)^i \cdot \frac{(n-1)!}{i!}$$

$$= (-1)^n + n \cdot dp[n-1]$$

一 容斥与dp和生成函数

对n个点有标号的有向无环图(可以不连通)进行计数.
答案对998244353取模.

subtask1: $n \le 10000$

subtask2: $n \le 100000$

一 容斥与dp和生成函数

设f(x)表示大小为x时的答案. 由于无环, 因此一定有入/出度为0的点, 可以将全图分为两个部分, 钦定一部分强制入度为零, 另一部分是任意的DAG, 变为子问题f(n - i).

讨论两部分之间的连边方式, 只能是入度为零的部分向另一部分任意连边, 有 $2^{i(n-i)}$种方式. 那么可以枚举强制为0的部分的大小. 由于无法保证另一部分一定不为0, 所以需要容斥, 式子如下:

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

这样我们就可以做到O(n²)的优秀复杂度. 继续优化:

一 容斥与dp和生成函数

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

让我们深情凝视这个式子, 发现它十分像多项式求逆的形式: f = f * g

于是我们, 先拆开组合数:

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

$$\frac{f(n)}{n!} = \sum_{i=1}^{n} \frac{f(n-i)}{(n-i)!} \cdot \frac{(-1)^{i-1} \cdot 2^{i^2}}{i!} \times 2^{\mathbf{in}}$$

我们的到了一个非常优美的形式, 但是最后的2^{in}部分十分麻烦.

容斥与dp和生成函数

$$\frac{f(n)}{n!}=\sum_{i=1}^{n}\frac{f(n-i)}{(n-i)!}\cdot\frac{(-1)^{i-1}\cdot 2^{i^2}}{i!}\times 2^{\mathbf{in}}$$

我们想到可以利用完全平方公式$2\cdot n\cdot i=n^2+i^2-(n-i)^2$.

并且2在mod 998244353的意义下有二次剩余, 于是我们有:

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

$$\frac{f(n)}{n!}=\sum_{i=1}^{n}\frac{f(n-i)}{(n-i)!}\cdot\frac{(-1)^{i-1}\cdot 2^{i^2}}{i!}\times \sqrt{2}^{n^2+i^2-(n-i)^2}$$

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

容斥与dp和生成函数

于是我们令H(x)表示 $\frac{f(x)}{x!\sqrt{2}^{\mathbf{x^2}}}$ 的生成函数, G(x)表示$\frac{(-1)^{x-1}}{\sqrt{2}^{\mathbf{x^2}}\cdot x!}$的生成函数. 我是~~不是跑题了~~

我们有: $H*G=\sum_{i=0}^{x}x^i(\sum_{j+k=i}f(j)\cdot g(k))$

因为我们要使得h(1)=g(1)h(0), 并且$h(x)=\frac{f(n)}{n!\sqrt{2}^{\mathbf{n^2}}}=\sum_{i=1}^{n}h(x-i)\cdot g(x)$的下标是从1开始的, 于是我们令g(0)=0, h(0)=1.

那么, 我们就有: $H*G+1=H$

$$H=\frac{1}{1-G}$$

一 容斥与数论

Codeforces 585E

给定大小为n的数列$\{a_i\}$, 求这样选数的方案数：先从其中挑出一个GCD不为1的集合，然后再选一个不属于该集合，且与该集合内任意一个数互质的数.

$n \le 10^5$, $a_i \le 10^7$

—— 容斥与数论 Codeforces 585E

先只考虑选择一些数字使得GCD≠1，容易想到枚举GCD.

先$O(n\ ln\ n)$地枚举g的倍数，得到cnt[g]表示数列中数字为g的倍数的个数.

那么含有公因数g的方案数为 $2^{cnt[g]}-1$，考虑容斥.

由于GCD≠1, 因此每个方案中至少有一个质数作为他们的公因数. 那么就相当于: 对于每个质数, 将[至少拥有这个质数]设为条件, 取所有条件的并集.

而这个的本质其实就是莫比乌斯函数μ. 那么总方案数就是:

$$sum=\sum_{g=2}^{Range}-\mu(g)*(2^{cnt[g]}-1)$$

容斥与数论 Codeforces 585E

$$sum = \sum_{g=2}^{Range} -\mu(g) * (2^{cnt[g]} - 1)$$

接下来考虑和另一个数组合. 发现实际上可以套在刚刚的容斥里. 我们先考虑一个数有多少集合不可以与之配对, 再用所有方案减去它的. 根据刚刚的容斥, 我们有:

$$\sum_{g|a_i}^{Range} -\mu(g) * (2^{cnt[g]} - 1)$$

那么我们换个角度, 对于一个g, 它的倍数cnt[g]都会对它产生-1的贡献, 就有:

$$Ans = \sum_{g=2}^{Range} -\mu(g) \cdot (2^{cnt[g]} - 1) \cdot (n - cnt[g])$$

一 容斥原理和各种技巧

LOJ2541 [PKUWC2018] 猎人杀

有n个球, 每个球有权值w_i, 每个时刻选择一个还存在的球扔掉, 每个球i被选中的概率是 $\frac{w_i}{\sum\limits_{v \in T} w_v}$ (T表示剩下的球的集合), 问1号球最后被扔掉的概率是多少.

$\sum w_i \le 1e5$

容斥原理和各种技巧

概率是动态的, 难以列出方便的式子. 尝试去掉不能选选过的球的限制. 可以理解为如果选中了选过的球就重新选择, 直到选到没选过的球. 简单证明一下, 设S表示全集, T表示扔掉的球的集合, 那么当前选中一个球x的概率就是:

$$\sum_{i=0}^{\infty}\left(\frac{w_T}{w_S}\right)^i \cdot \frac{w_x}{w_S} = \frac{1}{1-\frac{w_T}{w_S}} \cdot \frac{w_x}{w_S} = \frac{w_x}{w_S - w_T}$$

我们要求1号球最后被扔掉的概率, 我们可以理解为恰好有0个球比1号先扔掉. 那么就可以容斥了.

— 容斥原理和各种技巧

我们先设集合K中的球比1号先被扔掉. 那么概率为:

$$\sum_{i=0}^{\infty}(1-\frac{w_K+w_1}{w_S})^i \cdot \frac{w_1}{w_S} = \frac{w_1}{w_K+w_1}$$

于是:

$$Ans = \sum_{K \subseteq S}(-1)^k \cdot \frac{w_1}{w_K+w_1}$$

继续优化. 注意到题目中还有一个条件: $\Sigma w_i \le 1e5$. 我们发现，对于同样的权值的w_K, 若不考虑系数, 它们的贡献都是相同的, 而我们只有1e5个不同的w_K.

一 容斥原理和各种技巧

那么我们只要对于一个相同的w_K, 算出他们的容斥系数之和即可. 可以利用生成函数优化. 容斥系数与集合大小有关, 那么集合中的每个元素对系数产生-1的贡献, 那么它的生成函数就是: $\prod_{i=2}^{n}(1-x^{w_i})$

考虑用分治FFT优化.

我们把[l, r]拆成[l, mid], [mid + 1, r]分治下去, 再把它们得到的结果卷起来. 分治过程中, 每个区间记一个最高次数, 卷到最高次数即可. 这个过程类似于线段树.

记得回收空间.

一 容斥原理

容斥原理的应用十分广泛, 但题目中往往一带而过. 它方便了很多容易算重的过程. 当'恰好'难以计算时, 我们往往可以用'至多', '至少'取容斥.

当权值不大时, 我们可以考虑通过dp递推或者生成函数去求容斥系数之和, 这点在后边也会体现出来.

一 什么是反演

假设有两个函数$f(x)$和$g(x)$满足

$$f(n) = \sum_k a_{n,k} g(k)$$

已知f求g的过程就叫做反演。

— 二项式反演

二项式反演的本质其实就是容斥, 它的定义如下:

如果有: $$f_n = \sum_{i=0}^{n} \binom{n}{i} g_i$$

那么就有: $$g_n = \sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} f_i$$

二项式反演的简单证明

其实直接往里面带就好了:

$$
\begin{aligned}
g_n &= \sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}f_i \\
&= \sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}\sum_{j=0}^{i}\binom{i}{j}g_j \\
&= \sum_{j=0}^{n}g_j\sum_{i=j}^{n}\binom{n}{i}\binom{i}{j}(-1)^{n-i} \\
&= \sum_{j=0}^{n}g_j\sum_{i=j}^{n}\binom{n}{j}\binom{n-j}{i-j}(-1)^{n-i}
\end{aligned}
$$

— 二项式反演的简单证明

$$
\begin{aligned}
&= \sum_{j=0}^{n} g_j \sum_{i=j}^{n} \binom{n}{j} \binom{n-j}{i-j} (-1)^{n-i} \\
&= \sum_{j=0}^{n} g_j (\binom{n}{j} \sum_{i=j}^{n} \binom{n-j}{i-j} (-1)^{n-i}) \\
&= \sum_{j=0}^{n} g_j (\binom{n}{j} \sum_{i=0}^{n-j} \binom{n-j}{i} (-1)^{n-j-i}) \\
&= \sum_{j=0}^{n} g_j (\binom{n}{j} (1-1)^{n-j}) \\
&= g_n
\end{aligned}
$$

一 二项式反演的简单应用

BZOJ3622 已经没有什么好害怕的了

给定两个序列$\{a_i\}$和$\{b_i\}$, 要求重新排列a_i后$a_i > b_i$的对数恰好为k的方案数. 对10^9+9取模.

$n \le 2 \cdot 10^3$, $k \le n$ 保证所有数互不相同.

二项式反演的简单应用

显然先对A, B排序. 求出cnt[i], 表示比a_i小的数有多少个. 考虑dp, 设$dp[i][j]$表示到第i个数, 其中$a_i > b_i$的对数为j的方案数, 转移显然有:

$$dp[i][j] = dp[i-1][j] + dp[i-1][j-1] \cdot max(0, cnt[i] - (j-1))$$

我们令$f[i] = dp[n][i] \cdot (n-i)!$, 表示至少i组符合条件的方案数, 设$g[i]$表示恰好i种的方案数, 那么就有:

$$f[x] = \sum_{i=x}^{n} g[i]\binom{i}{x}$$

二项式反演得: $g[x] = \sum_{i=x}^{n}(-1)^i\binom{i}{x} \cdot f[i]$

二项式反演的应用

LOJ2527 [HAOI2018] 染色

为了报答小 C 的苹果, 小 G 打算送给热爱美术的小 C 一块画布, 这块画布可 以抽象为一个长度为 N 的序列, 每个位置都可以被染成 M 种颜色中的某一种.

然而小 C 只关心序列的 N 个位置中出现次数恰好为 S 的颜色种数, 如果恰 好出现了 S 次的颜色有 K 种, 则小 C 会产生 W_k 的愉悦度.

小 C 希望知道对于所有可能的染色方案, 他能获得的愉悦度的和对 1004535809 取模的结果是多少.

二项式反演的应用

设h[i]表示有i种颜色出现了S次的方案数，并且剩下的位置空着，那么就有:$h[i]=h[i-1]\cdot\binom{n-(i-1)S}{S}$

那么:

$$h[i]=\binom{n}{S}\cdot\binom{n-S}{S}\cdots\binom{n-(i-1)S}{S}=\frac{n!}{(s!)^i\cdot(n-iS)!}$$

设f[i]表示填满的情况下至少i种颜色出现了S次的方案数，就有:$f[i]=\binom{m}{i}\cdot h[i]\cdot(m-i)^{n-iS}$

设g[i]表示填满的情况下恰好i种颜色出现了S次的方案数，就有:$f[i]=\sum_{j=i}^{lim}\binom{j}{i}g[j]$

二项式反演的应用

令$Lim = \lfloor \frac{m}{S} \rfloor$, 二项式反演得到:

$g[x] = \sum_{i=x}^{Lim} (-1)^{i-x} \binom{i}{x} f[i]$

最终得到:

$$Ans = \sum_{i=0}^{Lim} g[i] w_i = \sum_{i=0}^{Lim} \sum_{j=i}^{Lim} (-1)^{j-i} \binom{j}{i} f[j]$$

$$= \sum_{i=0}^{Lim} i! \sum_{j=i}^{Lim} p[n-j+i] \cdot f[j]$$

其中$p[i] = \frac{(-1)^i}{i!}$. 这样就可以FFT了.

一 最值反演

定义:

给定集合S，设max(S)为S中的最大值，min(S)为S中的最小值，则我们有一个式子:

$$Max(S)=\sum_{T\subseteq S}(-1)^{|T|-1}Min(T)$$

一 最值反演

证明:

我们考虑构造一个容斥系数$f(x)$, 使得 $max(S) = \sum_{T\subseteq S} f(|T|)min(T)$

我们需要它剩下来的是最大值, 考虑第x+1大的元素会

被统计到的贡献 $[x == 0] = \sum_{i=0}^{x} C_x^i \times f(i+1)$

二项式反演一下: $f(x+1) = \sum_{i=0}^{x}(-1)^{x-i}C_x^i[i == 0]$

— 扩展最值反演

定义: $$\max_k\{S\}=\sum_{T\subseteq S}(-1)^{|T|-k}\binom{|T|-1}{k-1}Min\{T\}$$

证明同理

— 最值反演简单应用

51NOD 1355 斐波那契的最小公倍数

题目大意:

给 定 大 小 为 n 的 数 列 $\{a_i\}$, 求 $lcm\{fib(a_1), fib(a_2), fib(a_3) \cdots, fib(a_n)\}$.

$n \le 5 \cdot 10^4, a_i \le 10^6$

— 最值反演简单应用

这是一道十分经典的题. 先开始化式子:

$$
\begin{aligned}
lcm\{fib_S\} &= \prod_{p \in prime} p^{\max\limits_{i=1}^{n} r_i} \\
&= \prod_{p \in prime} p^{\sum\limits_{T \subseteq S} (-1)^{|T|-1} min\{T_{r_i}\}} \\
&= \prod_{T \subseteq S} gcd(fib_T)^{(-1)^{|T|-1}} \\
&= \prod_{T \subseteq S} fib_{gcd(T)}^{(-1)^{|T|-1}}
\end{aligned}
$$

最值反演简单应用

$$= \prod_{T \subseteq S} fib_{gcd(T)}^{(-1)^{|T|-1}}$$

令$f = g * 1$, 即$f(n) = \prod_{d|n} g(d)$:

$$Orignal = \prod_{T \subseteq S} \left(\prod_{d|gcd(T)} g(d) \right)^{(-1)^{|T|-1}}$$

$$= \prod_{d=2}^{Max} g(d)^{\sum_{T \subseteq S, T \neq \emptyset, d|T} (-1)^{|T|-1}}$$

一 最值反演简单应用

$$= \prod_{d=2}^{Max} g(d)^{\sum_{T\subseteq S, T\neq\emptyset, d|T} (-1)^{|T|-1}}$$

考虑g(d)指数的实际含义: $\sum_{T\subseteq S, T\neq\emptyset, d|T} (-1)^{|T|-1}$

发现这就是容斥的式子，那么它就等价于:

$$\sum_{T\subseteq S, T\neq\emptyset, d|T} (-1)^{|T|-1} = \begin{cases} 1 & \exists a_i, d|a_i \\ 0 & otherwise \end{cases}$$

最值反演简单应用

$$\sum_{T\subseteq S,T\neq\emptyset,d|T}(-1)^{|T|-1}=\begin{cases}1 & \exists a_i, d|a_i\\ 0 & otherwise\end{cases}$$

因此:

$$Original=\prod_{\exists a_i,d|a_i} g(d)$$

至于哪些d可行，可以O($n\ln n$)筛出来. 至于如何得到g，可以用ln筛.

— 扩展最值反演与期望

LG4707 重返现世

题目描述:

有n个球. 每次操作有$\frac{p_a}{\sum_{i=1}^{n} p_i}$选择一个球a, 求选中k个球的期望操作次数.

令$m = \sum p_i, n \leq 10^3, m \leq 10^4, |n-k| \leq 10$

扩展最值反演与期望

可以想到Min-Max容斥，发现题目要求的是选中k个球的时间，也就是每种方案中，选时间第k小的那个. 实际上就是选中时间第$n-k+1$大的.

直接给式子:

$$\max_k\{S\}=\sum_{T\subseteq S}(-1)^{|T|-k}\binom{|T|-1}{k-1}Min\{T\}$$

$$Min\{T\}=\sum_{i=0}^{\infty}(1-\frac{p_T}{m})^i\cdot\frac{p_T}{m}\cdot i$$

$$Min\{T\}=\frac{m}{p_T}$$

但复杂度还是2^n的. 发现T很多，而权值和很小，考虑(PKU2018 猎人杀)的套路，也就是枚举权值，计算每种权值前的系数和. 如果f(i)表示权值和为i前的系数和，那么答案就是: $Ans=\sum_{i=1}^{m}f[i]\cdot\frac{m}{i}$

扩展最值反演与期望

直接生成函数不好搞，不过数据范围提醒我们可以dp，因为容斥系数可以直接乘-1，而组合数可以递推. 尝试全部压进状态里.

设$dp[i][j][k]$表示考虑到第i个球，选出集合的权值和为j，其中组合数下标为k-1的容斥系数和，为了弄清转移的本质，先写出dp的实际意义，设$Cnt[i][j]$表示考虑到当前dp考虑到的球之前，集合大小为i，权值和为j的集合个数:

$$dp[i][j][k] = \sum_{t=0}^{j}(-1)^{t-k}\binom{t-1}{k-1}Cnt[t][j]$$

扩展最值反演与期望

第i个球有两种情况:

1.不塞进原来的任意集合，直接从$dp[i-1][j][k]$转移

2.强制塞进原来的集合，可以通过组合数递推得到:

那么我们就得到了一个非常优美的转移:

$$dp[i][j][k]=dp[i-1][j][k]+dp[i-1][j-p_i][k-1]-dp[i-1][j-p_i][k]$$

注意边界问题.

斯特林反演

定义:

$$f(n)=\sum_{k=0}^{n}\left\{ n \atop k \right\} g(k) \Longleftrightarrow g(n)=\sum_{k=0}^{n}(-1)^{n-k}\left[n \atop k \right] f(k)$$

—— 斯特林反演的应用

雅礼集训18.1.16方阵

题目大意:

给定n*m的矩阵, 可以填(1, C)中的数, 求任意两行, 任意两列均不相等的方案数.$(n, m \le 5000)$

— 斯特林反演的应用

对这种行列都有限制的题我们可以先只考虑一边.

那么有m列只让行之间互不等价的方案数, 设为g(m).那么就有 $g(m) = (C^m)^n$

我们设$f(m)$表示行和列都分别互不等价的情况下，m列的矩形的方案数，也就是我们要的答案。

$$g(m) = \sum_{i=0}^{m} \left\{ {m \atop i} \right\} f(i)$$

这个式子的意义就是我们枚举m列分成了i个互不等价的集合，再将这m列分配到这些集合中去。

由斯特林反演得到：

$$f(m) = \sum_{i=0}^{m} (-1)^{m-i} \left[{m \atop i} \right] g(i)$$

于是我们就可以$O(n^2)$计算了。

一 单位根反演的前置知识

0.1 单位根的定义

在复平面中，作一个单位圆，将其圆弧分为n个部分，以这些n等分点为终点作出n个向量. 其中，我们称幅角最小的那个向量为n次单位根，记为$\boldsymbol{w_n^1}$

0.2 单位根的各种性质

性质一： $\omega_n^k = \omega_n^{k-1} \cdot \omega_n^1$

性质二： 若 2 | n , 则有$\omega_n^i = -\omega_n^{i+\frac{n}{2}}$

性质三： $\omega_{kn}^{ki} = \omega_n^i$

单位根反演的前置知识

$$\forall k, [n \mid k] = \frac{1}{n}\sum_{i=0}^{n-1}\omega_n^{ik}$$

证明:

若$n \mid k$，则有

$$Ans = \frac{1}{n}\sum_{i=0}^{n-1}\omega_n^{ink'} = \frac{1}{n}\sum_{i=0}^{n-1}1 = 1$$

否则，这玩意是个等比数列求和，根据高中知识，我们有

$$Ans = \frac{1}{n}\frac{\omega_n^0 - \omega_n^{nk}}{1-\omega_n^k} = 0$$

得证。

单位根反演

单位根反演可以O($k \cdot \alpha$)内快速求出一个多项式中系数是k的倍数之和.

其中α为f(x)的计算速度.

对于$f(x) = \sum_{i=0}^{n} a_i x^i$, 要求 $\sum_{i=0}^{n} a_i [k|i]$

那么

$$\frac{\sum_{i=0}^{N-1} f(w_N^i)}{N} = \frac{\sum_{i=0}^{n} a_i \sum_{j=0}^{N-1} w_N^{ij}}{N}$$
$$= \sum_{i=0}^{n} a_i [N \mid i]$$

于是, 我们要求的就是$\frac{\sum_{i=0}^{N-1} f(w_N^i)}{N}$

— 单位根反演的简单应用

BZOJ3328 PYXFIB

题目大意:

求$Ans = \sum_{k|i}^{n} fib(i) \cdot \binom{n}{i}$

$(1 \le n \le 10^{18} 1 \le k \le 20000, 1 \le p \le 10^9$ p是素数且$p \bmod k = 1)$

单位根反演的简单应用

设$f(x) = \sum_{i=0}^{n} \binom{n}{i} Fib(i) \cdot x^i$, 数列A表示f(x)的系数.则 Ans = $\sum_{i=0}^{n} a_i [k|i]$.

Fib 不 带 指 数 , 十 分 不 快 . 考 虑 到 矩 幂 后 的 Fib 有 指 数 , 于 是 令 $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$, 则 有 $f(x) = \sum_{i=0}^{n} \binom{n}{i} A^i x^i \cdot I^{n-i}$,

有矩阵在式子中没有关系, 我们令矩阵乘数表示取矩阵右上角与之相乘.

由于与I 相乘的矩阵乘法满足交换律, 于是仍可以用二项式定理,

那么$f(x) = (Ax + I)^n$. $Ans = \frac{\sum_{i=0}^{k-1} f(w_k^i)}{k}$

总结

反演是一种强大的工具. 需要在比赛中灵活地使用.

常常用到的技巧就是改变枚举方向, 比如说: 数论中的枚举约数转枚举倍数.

还有改变枚举顺序, 将较为好算的部分放在一起.

容易算重时, 可以往容斥, 反演方向想.

容斥中出现的2次幂枚举, 可以利用生成函数或者dp优化.

THANK YOU

Lime 2019.7