组合计数选讲

JackF

2809811157@qq.com

January 2023

Will liuhengxi achieve LGM in grade 9?

By dXqwq, history, 10 days ago,

> As far as I know, he is in grade 9 now, which is one year younger than orzdevinwang.
>
> He is gaining rating in recent contests and previous LGMs in grade 9 are djq-cpp and orzdevinwang(comment below if you know more).
>
> Meanwhile I'm stuck between 2600~2700, hope I will become stronger next year o(>_<)o

+49 | dXqwq | 10 days ago | 5

Comments (5)

Write comment?

N_z__ — 10 days ago, # Edit | +34

As a classmate of liuhengxi,I think he will.

→ Reply

moonrise__ — 10 days ago, # ^ | +68

As a classmate of liuhengxi, I think he wants a girlfriend.

→ Reply

User_Carrot — 10 days ago, # ^ | +20

As a classmate and also a fan of liuhengxi, I think he will. And he wants a girlfriend btw.

→ Reply

noname 1

题目描述

求有多少个有序集合对 (A, B) 满足 $A \bigcup B = \{1, 2, \ldots, n\}$。

noname 1

一种做法，设 U 为全集，枚举集合 A 的大小，假设为 i。则集合 B 必须包含 $U - A$ 的所有元素，集合 A 中的元素包不包含都行。

noname 1

一种做法，设 U 为全集，枚举集合 A 的大小，假设为 i。则集合 B 必须包含 $U-A$ 的所有元素，集合 A 中的元素包不包含都行。

$$\sum_{i=0}^{n}\binom{n}{i}2^i=3^n$$

noname 1

一种做法，设 U 为全集，枚举集合 A 的大小，假设为 i。则集合 B 必须包含 $U-A$ 的所有元素，集合 A 中的元素包不包含都行。

$$\sum_{i=0}^{n}\binom{n}{i}2^i=3^n$$

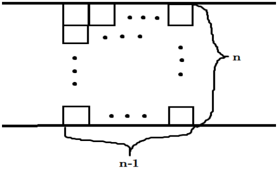
另一种做法，考虑任意一个元素 $e\in U$，e 的类型有三种：

- 只出现在 A 里。
- 只出现在 B 里。
- 在 A,B 中都出现。

n 个元素互相独立，因此方案数为 3^n。

noname 2

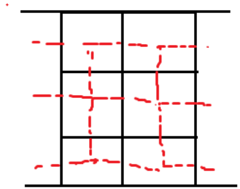
题目描述



如图，在河两岸之间有一些桥，这些桥形成了一个 $n \times (n-1)$ 的阵列，为每座桥确定出现/不出现状态，求能使得一个人从河的一边走到另一边的方案数。

noname 2

考虑其的对偶问题，假设有一艘船从左往右行驶，将船的行驶路径画出来，发现与原问题等价，而人能通过和船能通过不可能同时成立，因此恰好有 $2^{\text{bridge}-1}$ 种方案。

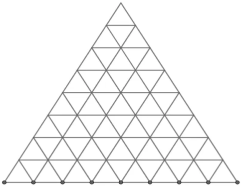


上图是 $n = 3$ 时的情况，其中红色虚线是船行驶路径。

noname 3

题目描述

求在边长为 n 的格点正三角形中选 3 个点满足 3 个点能构成正三角形的方案数。



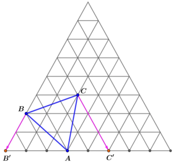
上图为 $n = 9$ 的情况。

Hint：考虑递推。

Hint：考虑递推。

设 S_n 为答案，分情况计算：

- 三个点都不在最后一行，方案数为 S_{n-1}。
- 两个点在最后一行，此时剩余一点确定，方案数为 $\binom{n}{2}$。
- 只有一个点在最后一行，考虑剩余两点在最后一行的 60° 和 120° 投影。



恰好对应了最后一行的不同的 3 个点，方案数为 $\binom{n}{3}$。

则 $S_n = S_{n-1} + \binom{n}{2} + \binom{n}{3} = S_{n-1} + \binom{n+1}{3} = 1 + \sum_{i=4}^{n+1} \binom{i}{3} = \binom{n+2}{4}$。

这部分比较基础。

正整数和的数目

将 n 个完全相同的元素分为 k 组，保证每组至少有一个元素，求方案数。

正整数和的数目

等同于 $x_1 + x_2 + \ldots + x_k = n$ 的正整数解的数量。
考虑用 $k-1$ 块板子插入到 n 个元素产生的 $n-1$ 个空里。
方案数即为 $\binom{n-1}{k-1}$。

有下界整数和的数目

求 $x_1 + x_2 + \ldots + x_k = n; \forall i \in [1, n], x_i \geq a_i$ 的整数解的数量。

有下界整数和的数目

考虑构造 $y_i = x_i - (a_i - 1)$，此时就转化成了前一种的形式。
答案即为 $\binom{n-\sum a_i+k-1}{k-1}$。

不相邻的排列

求从 $1 \sim n$ 这 n 个自然数中选 k 个，满足这 k 个数任何两个都不相邻的方案数。

不相邻的排列

设这 k 个数排完序是 $a_1, a_2, \ldots, a_k$。

不相邻的排列

设这 k 个数排完序是 $a_1, a_2, \ldots, a_k$。
令 $d_i = a_i - a_{i-1}$，则要满足的条件是

$$\begin{cases} d_1 > 0; \forall i \in [2, k], d_i > 1; \\ d_1 + d_2 + \ldots + d_k \le n \end{cases}$$

不相邻的排列

设这 k 个数排完序是 $a_1, a_2, \ldots, a_k$。
令 $d_i = a_i - a_{i-1}$，则要满足的条件是

$$\begin{cases} d_1 > 0; \forall i \in [2, k], d_i > 1; \\ d_1 + d_2 + \ldots + d_k \leq n \end{cases}$$

对于前一个限制，用前一题的方法，对于后一个限制，引入一个变量 $d_{k+1} \geq 0$ 并且把限制改为 $d_1 + d_2 + \ldots + d_{k+1} = n$ 即可。
答案为 $\binom{n-k+1}{k}$。

多重集的排列数

有 k 种不同元素，第 i 种有 a_i 个，求将这些元素排列的方案数。

多重集的排列数

如果不考虑相同元素，答案就是 $n!$。
一种有 x 个的相同元素会使得答案多乘上 $x!$。
因此答案为 $\frac{(a_1+a_2+...+a_k)!}{a_1!a_2!...a_k!}$。

容斥原理 - 定义

有 n 个集合 $S_1, S_2, \ldots, S_n$。

$$\left|\bigcup_{i=1}^{n} S_i\right| = \sum_{i} |S_i| - \sum_{i<j} |S_i \cap S_j| + \sum_{i<j<k} |S_i \cap S_j \cap S_k| - \cdots + (-1)^{m-1} \sum_{a_i<a_{i+1}} \left|\bigcap_{i=1}^{m} S_{a_i}\right| + \cdots + (-1)^{n-1} |S_1 \cap \cdots \cap S_n|$$

[HAOI2008] 硬币购物

题目描述

4 种面值的硬币，第 i 种的面值是 C_i。n 次询问，每次询问给出每种硬币的数量 D_i 和一个价格 S，问付款的方案数。 $n \le 10^3, S \le 10^5$。

[HAOI2008] 硬币购物

每次询问暴力做背包复杂度是 $O(4nS)$。

[HAOI2008] 硬币购物

每次询问暴力做背包复杂度是 $O(4nS)$。
抽象化题目，要求的即为 $\sum_{i=1}^{4} C_i x_i = S, x_i \leq D_i$ 的非负整数解的数量。
容斥上界，枚举哪些 $x_i > D_i$，然后将 S 减去 $C_i(D_i+1)$，剩下就是无上界的问题，也就是无限背包问题，这个提前预处理即可。
总时间复杂度 $O(4S + 2^4 n)$。

错位排列

求满足 $\forall i, P_i \neq i$ 的 $1 \sim n$ 的排列 P 的数量。

错位排列

一种方法是考虑递推。
设 D_n 表示 $1 \sim n$ 的错位排列数。
考虑两种情况：

- 前面全部错位，此时将 n 与前面任意一个进行交换仍然是错位排列。
- 前面有且仅有一个没有错位，此时将 n 与没有错位的交换产生一种新的错位排列。

则 $D_n = (n-1)(D_{n-1} + D_{n-2})$。

错位排列

一种方法是考虑递推。
设 D_n 表示 $1 \sim n$ 的错位排列数。
考虑两种情况:

- 前面全部错位，此时将 n 与前面任意一个进行交换仍然是错位排列。
- 前面有且仅有一个没有错位，此时将 n 与没有错位的交换产生一种新的错位排列。

则 $D_n = (n-1)(D_{n-1} + D_{n-2})$。
另一种方法就是使用容斥。
容斥至少有几个位置保持原位。
则

$$D_n = \sum_{k=0}^{n} (-1)^k \binom{n}{k} (n-k)! = n! \sum_{k=0}^{n} \frac{(-1)^k}{k!}$$

DAG 计数

题目描述

对 n 个点带标号的有向无环图进行计数。 $1 \le n \le 5000$。

DAG 计数

考虑拓扑排序的过程，由于入度为 0 的点之间不会有边，将入度为 0 的点去掉。
枚举入度为 0 的点的数量 i，然后这 i 个点与剩下 $n-i$ 个点之间的边的状态有 $2^{i(n-i)}$。注意到此时没法保证恰好有 i 个入度为 0 的点，所以需要容斥一下，将恰好转为至少。

$$f_n=\sum_{i=1}^{n}(-1)^{i-1}\binom{n}{i}f_{n-i}\times 2^{i(n-i)}$$

直接转移是 $O(n^2)$ 的。

LG6295 有标号 DAG 计数

题目描述

求 $n = 1, 2, \ldots, T$ 个点有标号弱连通 DAG 数量。弱连通图是指将所有的有向边替换为无向边后的图为连通图。

$1 \le T \le 10^5$。

上一题的升级版。
首先考虑怎么求弱连通，非弱连通的 DAG 必定由多个连通分量组合，即若 DAG 的 EGF 为 $F(x)$，非弱联通 DAG 的 EGF 为 $H(x)$，则 $H(x) = \ln F(x)$。

上一题的升级版。
首先考虑怎么求弱连通，非弱连通的 DAG 必定由多个连通分量组合，即若 DAG 的 EGF 为 $F(x)$，非弱联通 DAG 的 EGF 为 $H(x)$，则 $H(x) = \ln F(x)$。
因此要快速求出 DAG 的 EGF。注意到 $i(n-i) = \binom{n}{2} - \binom{i}{2} - \binom{n-i}{2}$，所以有

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

上一题的升级版。
首先考虑怎么求弱连通，非弱连通的 DAG 必定由多个连通分量组合，即若 DAG 的 EGF 为 $F(x)$，非弱联通 DAG 的 EGF 为 $H(x)$，则 $H(x) = \ln F(x)$。
因此要快速求出 DAG 的 EGF。注意到 $i(n-i) = \binom{n}{2} - \binom{i}{2} - \binom{n-i}{2}$，所以有

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

设

$$F(x) = \sum_{i=0}^{\infty} \frac{f_i}{i! 2^{\binom{i}{2}}}$$

$$G(x) = \sum_{i=1}^{\infty} \frac{(-1)^{i-1}}{i! 2^{\binom{i}{2}}}$$

则 $F(x) = F(x)G(x) + 1 \Rightarrow F(x) = \frac{1}{1-G(x)}$。
只需要多项式求逆、取对数函数即可。时间复杂度 $O(n \log n)$。

第二类斯特林数

$\left\{ {n \atop m} \right\}$ 表示将 n 个不同元素划分成 m 个相同的集合（不能有空集）的方案数。

第二类斯特林数

$\left\{ n \atop m \right\}$ 表示将 n 个不同元素划分成 m 个相同的集合（不能有空集）的方案数。
考虑递推，当加入第 n 个元素时，一共有两种情况：

- 加入前面的某一组，方案数为 $m \times \left\{ n-1 \atop m \right\}$。
- 自己为新的一组，方案数 $\left\{ n-1 \atop m-1 \right\}$。

故 $\left\{ n \atop m \right\} = m \times \left\{ n-1 \atop m \right\} + \left\{ n-1 \atop m-1 \right\}$，可以 $O(n^2)$ 递推计算。

第二类斯特林数

$\left\{{n \atop m}\right\}$ 表示将 n 个不同元素划分成 m 个相同的集合（不能有空集）的方案数。
考虑递推，当加入第 n 个元素时，一共有两种情况：

- 加入前面的某一组，方案数为 $m \times \left\{{n-1 \atop m}\right\}$。
- 自己为新的一组，方案数 $\left\{{n-1 \atop m-1}\right\}$。

故 $\left\{{n \atop m}\right\} = m \times \left\{{n-1 \atop m}\right\} + \left\{{n-1 \atop m-1}\right\}$，可以 $O(n^2)$ 递推计算。
另一种方法是使用容斥，枚举有多少个空集：

$$\left\{{n \atop m}\right\} = \frac{1}{m!}\sum_{i=0}^{m}(-1)^i\binom{m}{i}(m-i)^n$$

把组合数拆开来，有

$$\left\{{n \atop m}\right\} = \sum_{i=0}^{m}\frac{(-1)^i}{i!}\frac{(m-i)^n}{(m-i)!}$$

可以卷积 $O(n\log n)$ 算出一行。

普通幂转下降幂

Formula

$$m^n = \sum_{i=0}^{n} \left\{ {n \atop i} \right\} m^{\underline{i}}$$

普通幂转下降幂

Formula

$$m^n = \sum_{i=0}^{n} \left\{ {n \atop i} \right\} m^{\underline{i}}$$

考虑用组合意义证明。

普通幂转下降幂

Formula

$$m^n = \sum_{i=0}^{n} \left\{ {n \atop i} \right\} m^{\underline{i}}$$

考虑用组合意义证明。
m^n 可以看作将 n 个物品放入 m 个不同的盒子的方案数。
考虑枚举有几个非空盒子，从 m 个盒子中选出 i 个方案数为 $\binom{m}{i}$，然后将 n 个物品放入 m 个盒子的方案数为 $\left\{ {n \atop i} \right\}$，由于盒子有顺序，所以再乘上 $i!$。

普通幂转下降幂

Formula

$$m^n = \sum_{i=0}^{n} \left\{ {n \atop i} \right\} m^{\underline{i}}$$

考虑用组合意义证明。
m^n 可以看作将 n 个物品放入 m 个不同的盒子的方案数。
考虑枚举有几个非空盒子，从 m 个盒子中选出 i 个方案数为 $\binom{m}{i}$，然后将 n 个物品放入 m 个盒子的方案数为 $\left\{ {n \atop i} \right\}$，由于盒子有顺序，所以再乘上 $i!$。

$$\sum_{i=0}^{m} \left\{ {n \atop i} \right\} \binom{m}{i} i! = \sum_{i=0}^{m} \left\{ {n \atop i} \right\} m^{\underline{i}}$$

自然数 k 次幂和

求 $\sum_{i=0}^{n} i^k$。

自然数 k 次幂和

求 $\sum_{i=0}^{n} i^k$。

这是个非常经典的题，一种常见的方法是拉格朗日插值 $O(k)$ 解决。

自然数 k 次幂和

求 $\sum_{i=0}^{n} i^k$。

这是个非常经典的题，一种常见的方法是拉格朗日插值 $O(k)$ 解决。
下面给出一种用第二类斯特林数求的做法。

$$\begin{aligned}\sum_{i=0}^{n} i^k &= \sum_{i=0}^{n}\sum_{j=0}^{k}\left\{{k \atop j}\right\} j!\binom{i}{j} \\ &= \sum_{i=0}^{k}\left\{{k \atop i}\right\} i!\sum_{j=0}^{n}\binom{j}{i} \\ &= \sum_{i=0}^{k}\left\{{k \atop i}\right\} i!\binom{n+1}{i+1}\end{aligned}$$

CF1716F Bags with Balls

题目描述

有 n 个不同的盒子，每个盒子里有 m 个编号分别为 $1, 2, \ldots, m$ 的小球。现在要从每个盒子中恰好取出 1 个球，计算每种取法中，编号是奇数的小球个数的 k 次方和。

$1 \leq T \leq 5000, 1 \leq n, m \leq 998244352, 1 \leq k \leq 2000$。

CF1716F Bags with Balls

设 a 为 $1 \sim m$ 中奇数的数量，b 为 $1 \sim m$ 中偶数的数量。考虑生成函数，令 $F(x) = (ax + b)^n$，要求的即为

$$\text{Ans} = \sum_{i=1}^{n} [x^i]F(x) \times i^k$$

CF1716F Bags with Balls

设 a 为 $1 \sim m$ 中奇数的数量，b 为 $1 \sim m$ 中偶数的数量。考虑生成函数，令 $F(x) = (ax + b)^n$，要求的即为

$$\text{Ans} = \sum_{i=1}^{n} [x^i]F(x) \times i^k$$

由二项式定理，$[x^i]F(x) = \binom{n}{i} a^i b^{n-i}$。

CF1716F Bags with Balls

设 a 为 $1 \sim m$ 中奇数的数量，b 为 $1 \sim m$ 中偶数的数量。考虑生成函数，令 $F(x) = (ax+b)^n$，要求的即为

$$\text{Ans} = \sum_{i=1}^{n} [x^i]F(x) \times i^k$$

由二项式定理，$[x^i]F(x) = \binom{n}{i} a^i b^{n-i}$。

$$\begin{aligned}\text{Ans} &= \sum_{i=1}^{n} \binom{n}{i} a^i b^{n-i} \sum_{j=1}^{k} \left\{ {k \atop j} \right\} \binom{i}{j} j! \\ &= \sum_{j=1}^{k} \left\{ {k \atop j} \right\} j! \sum_{i=1}^{n} \binom{n}{i} \binom{i}{j} a^i b^{n-i} \\ &= \sum_{j=1}^{k} \left\{ {k \atop j} \right\} n^{\underline{j}} \sum_{i=1}^{n} \binom{n-j}{i-j} a^i b^{n-i}\end{aligned}$$

CF1716F Bags with Balls

考虑化简式子的后半部分。

$$\sum_{i=1}^{n}\binom{n-j}{i-j}a^i b^{n-i}$$

CF1716F Bags with Balls

考虑化简式子的后半部分。

$$\sum_{i=1}^{n} \binom{n-j}{i-j} a^i b^{n-i}$$

当 $i < j$ 时 $\binom{n-i}{i-j} = 0$，没有贡献，将式子改写为

$$\begin{aligned}
&\sum_{i=0}^{n-j} \binom{n-j}{i} a^{i+j} b^{n-i-j} \\
=&a^j \sum_{i=0}^{n-j} \binom{n-j}{i} a^i b^{n-j-i} \\
=&a^j (a+b)^{n-j}
\end{aligned}$$

CF1716F Bags with Balls

考虑化简式子的后半部分。

$$\sum_{i=1}^{n}\binom{n-j}{i-j}a^i b^{n-i}$$

当 $i<j$ 时 $\binom{n-i}{i-j}=0$，没有贡献，将式子改写为

$$\begin{aligned}&\sum_{i=0}^{n-j}\binom{n-j}{i}a^{i+j}b^{n-i-j}\\=&a^j\sum_{i=0}^{n-j}\binom{n-j}{i}a^i b^{n-j-i}\\=&a^j(a+b)^{n-j}\end{aligned}$$

第二类斯特林数可以 $O(k^2)$ 预处理出来，n 的下降幂可以 $O(k)$ 预处理，总时间复杂度 $O(k^2+Tk)$。

LG2791 幼儿园篮球题

题目描述

原题题面比较抽象，~~大家可以自己去看~~。

q 次询问，每次求

$$\frac{\sum_{i=0}^{k} i^L \binom{m}{i}\binom{n-m}{k-i}}{\binom{n}{k}}$$

$1 \le q \le 200, 1 \le L \le 2 \cdot 10^5, 1 \le m, k \le n \le 2 \cdot 10^7$。

LG2791 幼儿园篮球题

分母平凡，考虑把分子的 i^L 拆了，得到

$$\sum_i \binom{m}{i}\binom{n-m}{k-i}\sum_j \begin{Bmatrix} L \\ j \end{Bmatrix} i^{\underline{j}}$$

LG2791 幼儿园篮球题

分母平凡，考虑把分子的 i^L 拆了，得到

$$\sum_i \binom{m}{i}\binom{n-m}{k-i}\sum_j \left\{ L \atop j \right\} i^{\underline{j}}$$

交换求和号，将下降幂提取为组合数

$$\begin{aligned}
&\sum_j \left\{ L \atop j \right\} j! \sum_i \binom{m}{i}\binom{n-m}{k-i}\binom{i}{j} \\
=&\sum_j \left\{ L \atop j \right\} \binom{m}{j} j! \sum_i \binom{m-j}{i-j}\binom{n-m}{k-i} \\
=&\sum_j \left\{ L \atop j \right\} \binom{m}{j} j! \binom{n-j}{k-j}
\end{aligned}$$

LG2791 幼儿园篮球题

分母平凡，考虑把分子的 i^L 拆了，得到

$$\sum_i \binom{m}{i}\binom{n-m}{k-i}\sum_j \begin{Bmatrix} L \\ j \end{Bmatrix} i^{\underline{j}}$$

交换求和号，将下降幂提取为组合数

$$\begin{aligned}
&\sum_j \begin{Bmatrix} L \\ j \end{Bmatrix} j! \sum_i \binom{m}{i}\binom{n-m}{k-i}\binom{i}{j} \\
=&\sum_j \begin{Bmatrix} L \\ j \end{Bmatrix}\binom{m}{j} j! \sum_i \binom{m-j}{i-j}\binom{n-m}{k-i} \\
=&\sum_j \begin{Bmatrix} L \\ j \end{Bmatrix}\binom{m}{j} j! \binom{n-j}{k-j}
\end{aligned}$$

$O(L\log L)$ 预处理一行第二类斯特林数，每次查询 $O(L)$ 计算，总时间复杂度 $O(L\log L+qL)$。

LG4827 [国家集训队] Crash 的文明世界

题目描述

给定一棵 n 个节点的树，对于每个点 i，求出

$$S(i)=\sum_{j=1}^{n}\mathrm{dist}(i,j)^k$$

其中 $\mathrm{dist}(x,y)$ 是 x 和 y 的树上距离。
$1\le n\le 5\cdot 10^4, 1\le k\le 150$。

LG4827 [国家集训队] Crash 的文明世界

将 $\text{dist}(i,j)^k$ 拆开。

$\sum_{j=1}^{n} \text{dist}(i,j)^k = \sum_{t=0}^{k} {k \brace t} t! \sum_{j=1}^{n} \binom{\text{dist}(i,j)}{t}$。

LG4827 [国家集训队] Crash 的文明世界

将 $\text{dist}(i,j)^k$ 拆开。
$\sum_{j=1}^{n}\text{dist}(i,j)^k=\sum_{t=0}^{k}\left\{{k \atop t}\right\}t!\sum_{j=1}^{n}\binom{\text{dist}(i,j)}{t}$。
考虑 $\binom{\text{dist}(i,j)}{t}=\binom{\text{dist}(i,j)-1}{t}+\binom{\text{dist}(i,j)-1}{t-1}$。
设 $dp_{x,i}$ 表示 $\sum_{y\in\text{subtree}_x}\binom{\text{dist}(x,y)}{i}$。
那么有 $dp_{x,i}=\sum_{y\in\text{son}_x}dp_{y,i}+dp_{y,i-1}$。

LG4827 [国家集训队] Crash 的文明世界

将 $\text{dist}(i,j)^k$ 拆开。
$\sum_{j=1}^{n}\text{dist}(i,j)^k=\sum_{t=0}^{k}\left\{{k \atop t}\right\}t!\sum_{j=1}^{n}\binom{\text{dist}(i,j)}{t}$。
考虑 $\binom{\text{dist}(i,j)}{t}=\binom{\text{dist}(i,j)-1}{t}+\binom{\text{dist}(i,j)-1}{t-1}$。
设 $dp_{x,i}$ 表示 $\sum_{y\in\text{subtree}_x}\binom{\text{dist}(x,y)}{i}$。
那么有 $dp_{x,i}=\sum_{y\in\text{son}_x}dp_{y,i}+dp_{y,i-1}$。
注意到这样只能算出根的答案，那么算完之后再进行一次换根 dp 即可。

时间复杂度 $O(nk)$。

二项式反演 - 定义

有两种形式:

$$f_n=\sum_{i=0}^{n}\binom{n}{i}g_i \Longleftrightarrow g_n=\sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}f_i$$

$$f_n=\sum_{i=n}^{m}\binom{i}{n}g_i \Longleftrightarrow g_n=\sum_{i=n}^{m}(-1)^{i-n}\binom{i}{n}f_i$$

下面来证明一下。

二项式反演 - 证明

这边挑选形式二进行证明，形式一的证明可以类似得到。
将形式二的右式代入左式：

$$
\begin{aligned}
f_n &= \sum_{i=n}^{m} \binom{i}{n} \sum_{j=i}^{m} (-1)^{j-i} \binom{j}{i} f_j \\
&= \sum_{j=n}^{m} f_j \sum_{i=n}^{j} (-1)^{j-i} \binom{i}{n} \binom{j}{i} \\
&= \sum_{j=n}^{m} \binom{j}{n} f_j \sum_{t=0}^{j-n} \binom{j-n}{t} (-1)^t \\
&= \sum_{j=n}^{m} \binom{j}{n} f_j [j=n] \\
&= f_n
\end{aligned}
$$

二项式反演 - 证明

下面来介绍另一种证明方法，以形式一为例。
设 f 和 g 的 EGF 是 $F(x), G(x)$，则

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

可以改写为

$$F(x) = e^x G(x)$$

那么

$$G(x) = e^{-x} F(x)$$

展开 e^{-x} 即可得到右式。

bzoj2839 集合计数

题目描述

一个有 n 个元素的集合有 2^n 个不同的子集，现在要从这 2^n 个集合中挑出至少一个集合，使得它们的交集大小恰好为 k，求方案数。

$1 \leq n \leq 10^6, 0 \leq k \leq n$。

bzoj2839 集合计数

若钦定 i 个交集元素，包含这 i 个元素的集合有 2^{n-i} 个。每个集合都可选或者不选，但不能全部不选。因此方案数为 $\binom{n}{i}(2^{2^{n-i}}-1)$。

bzoj2839 集合计数

若钦定 i 个交集元素，包含这 i 个元素的集合有 2^{n-i} 个。每个集合都可选或者不选，但不能全部不选。因此方案数为 $\binom{n}{i}(2^{2^{n-i}}-1)$。
设 f_i 为钦定交集元素为某 i 个的方案数，g_i 表示交集元素恰好为 i 个的方案数，则有 $f_i=\binom{n}{i}(2^{2^{n-i}}-1)=\sum_{k=i}^{n}\binom{k}{i}g_k$。

bzoj2839 集合计数

若钦定 i 个交集元素，包含这 i 个元素的集合有 2^{n-i} 个。每个集合都可选或者不选，但不能全部不选。因此方案数为 $\binom{n}{i}(2^{2^{n-i}}-1)$。
设 f_i 为钦定交集元素为某 i 个的方案数，g_i 表示交集元素恰好为 i 个的方案数，则有 $f_i=\binom{n}{i}(2^{2^{n-i}}-1)=\sum_{k=i}^{n}\binom{k}{i}g_k$。
也就是形式二，那么根据二项式反演得出
$g_i=\sum_{k=i}^{n}(-1)^{k-i}\binom{k}{i}f_k=\sum_{k=i}^{n}(-1)^{k-i}\binom{k}{i}\binom{n}{k}(2^{2^{n-k}}-1)$。
时间复杂度 $O(n)$。

LG4859 已经没有什么好害怕的了

题目描述

给定两个长度为 n 的序列 A, B，保证这 $2n$ 个数互不相同。将 A 序列中的数与 B 序列中的数两两配对，求出 $A > B$ 的对数比 $A < B$ 的对数恰好多 k 的配对方案数。
$1 \le n \le 2000, 0 \le k \le n$。

LG4859 已经没有什么好害怕的了

显然 $n+k$ 为奇数无解，$A>B$ 的对数为 $\frac{n+k}{2}$。

LG4859 已经没有什么好害怕的了

显然 $n+k$ 为奇数无解，$A>B$ 的对数为 $\frac{n+k}{2}$。
考虑将恰好转化为钦定，再用二项式反演来处理。将 A 和 B 从小到大排序，设 $dp_{i,j}$ 表示考虑 A 的前 i 个数，钦定 j 对 $A>B$ 的配对方案数。
转移显然，$dp_{i,j}=dp_{i-1,j}+dp_{i-1,j-1}\times(\sum_{j=1}^{n}[B_j<A_i]-(j-1))$。

LG4859 已经没有什么好害怕的了

显然 $n+k$ 为奇数无解，$A>B$ 的对数为 $\frac{n+k}{2}$。
考虑将恰好转化为钦定，再用二项式反演来处理。将 A 和 B 从小到大排序，设 $dp_{i,j}$ 表示考虑 A 的前 i 个数，钦定 j 对 $A>B$ 的配对方案数。
转移显然，$dp_{i,j}=dp_{i-1,j}+dp_{i-1,j-1}\times(\sum_{j=1}^{n}[B_j<A_i]-(j-1))$。
设 f_i 表示钦定 i 对 $A>B$ 的方案数，g_i 表示恰好 i 对 $A>B$ 的方案数，则 $f_i=(n-i)!\times dp_{n,i}=\sum_{k=i}^{n}\binom{k}{i}g_k$。
所以 $g_i=\sum_{k=i}^{n}(-1)^{k-i}\binom{k}{i}f_k$，最后将 $\frac{n+k}{2}$ 代入即可。
时间复杂度 $O(n^2)$。

[JSOI2011] 分特产

题目描述

有 n 个人和 m 种物品，第 i 种物品有 a_i 种，同种物品直接没有区别。要求将物品分给这 n 个人使得每个人至少分到一个物品，求方案数。
$1 \leq n, m \leq a_i \leq 1000$。

[JSOI2011] 分特产

考虑将每个人至少分到一个物品转化为恰好 0 个人没有分到物品，然后用二项式反演来处理。

[JSOI2011] 分特产

考虑将每个人至少分到一个物品转化为恰好 0 个人没有分到物品，然后用二项式反演来处理。

设 f_i 表示钦定 i 个人没有分到物品的方案数，g_i 表示恰好 i 个人没有分到物品的方案数。计算 f_i 时，第 j 种物品的方案数即 $x_1 + x_2 + \ldots + x_{n-i} = a_j$ 的非负解组数，用插板法求出为 $\binom{n-i+a_j-1}{a_j-1}$。

那么 $f_i = \prod_{j=1}^{m} \binom{n}{i}\binom{n-i+a_j-1}{a_j-1} = \sum_{k=i}^{n} \binom{k}{i} g_k$。

[JSOI2011] 分特产

考虑将每个人至少分到一个物品转化为恰好 0 个人没有分到物品，然后用二项式反演来处理。
设 f_i 表示钦定 i 个人没有分到物品的方案数，g_i 表示恰好 i 个人没有分到物品的方案数。计算 f_i 时，第 j 种物品的方案数即 $x_1 + x_2 + \ldots + x_{n-i} = a_j$ 的非负解组数，用插板法求出为 $\binom{n-i+a_j-1}{a_j-1}$。
那么 $f_i = \prod_{j=1}^{m} \binom{n}{i}\binom{n-i+a_j-1}{a_j-1} = \sum_{k=i}^{n} \binom{k}{i} g_k$。
二项式反演一下，$g_0 = \sum_{i=0}^{n} (-1)^i \binom{n}{i} \prod_{j=1}^{m} \binom{n-i+a_j-1}{a_j-1}$。
时间复杂度 $O(n^2 + nm)$。

[NOI Online #2 提高组] 游戏

题目描述

给定一棵包含 $n = 2m$ 个点的有根树，根节点是 1 号点，小 A(apy) 和小 B(bsf) 在树上玩游戏。
初始时两人各拥有 m 个点，游戏一共有 m 个回合，每个回合两人都需要选出一个自己拥有且之前未选过的点，如果对手的点在自己的点的子树内，则该回合自己获胜；若自己的点在对方的点的子树内，该回合对方获胜；其他情况视为平局。
对于 $k = 0, 1, 2, \ldots, m$，计算出非平局回合数为 k 的情况数。
$n \leq 5000$。

[NOI Online #2 提高组] 游戏

设 A 拥有的点为 $p_1, p_2, \ldots, p_m$，B 拥有的点为 $q_1, q_2, \ldots, q_m$，题目等价于求有多少个排列 u 满足恰好存在 k 对 (p_i, q_{u_i}) 是祖先后代关系。

[NOI Online #2 提高组] 游戏

设 A 拥有的点为 $p_1, p_2, \ldots, p_m$，B 拥有的点为 $q_1, q_2, \ldots, q_m$，题目等价于求有多少个排列 u 满足恰好存在 k 对 (p_i, q_{u_i}) 是祖先后代关系。

将恰好转化为至少，考虑设 f_k 表示钦定 k 对必须匹配的点，剩余随意排列的方案数。可以通过二项式反演得到恰好 k 对必须匹配的方案数。

[NOI Online #2 提高组] 游戏

设 A 拥有的点为 $p_1, p_2, \ldots, p_m$，B 拥有的点为 $q_1, q_2, \ldots, q_m$，题目等价于求有多少个排列 u 满足恰好存在 k 对 (p_i, q_{u_i}) 是祖先后代关系。

将恰好转化为至少，考虑设 f_k 表示钦定 k 对必须匹配的点，剩余随意排列的方案数。可以通过二项式反演得到恰好 k 对必须匹配的方案数。

设 $dp_{x,i}$ 表示 x 的子树内钦定了 i 对要匹配的点的方案数。先做一个简单的树形背包合并，再考虑加入 x 这个节点的匹配情况。如果 x 属于 A，$dp_{x,i+1} \leftarrow dp_{x,i} \times (\sum_{y \in \text{subtree}_x} [y \in B]) - i)$。$x$ 属于 B 也同理。最后 $f_i = dp_{1,i} \times (\frac{n}{2} - i)!$。

时间复杂度 $O(n^2)$。

下面是一些杂题。

LG4389 符公主的背包

题目描述

符公主有 n 种商品，每种商品体积为 v_i，都有无限件。
给定 m，对于 $s \in [1, m]$，求出用这些商品恰好装 s 体积的方案数。
$1 \le n, m \le 10^5, 1 \le v_i \le m$。

LG4389 符公主的背包

考虑用生成函数表示。

LG4389 符公主的背包

考虑用生成函数表示。
已知 $\sum_{i\geq 0} x^{iv} = \frac{1}{1-x^v}$，那么要求的就是 $[x^s]\prod \frac{1}{1-x^{v_i}}$。

LG4389 符公主的背包

考虑用生成函数表示。
已知 $\sum_{i\geq 0} x^{iv} = \frac{1}{1-x^v}$，那么要求的就是 $[x^s]\prod \frac{1}{1-x^{v_i}}$。
设 $F(x) = \prod \frac{1}{1-x^{v_i}}$，对 F 取对数：$\ln F = \sum_i \ln \frac{1}{1-x^{v_i}}$。那么现在的问题就是求出 $\ln \frac{1}{1-x^k}$。

LG4389 付公主的背包

考虑用生成函数表示。

已知 $\sum_{i\geq 0} x^{iv} = \frac{1}{1-x^v}$，那么要求的就是 $[x^s]\prod \frac{1}{1-x^{v_i}}$。

设 $F(x) = \prod \frac{1}{1-x^{v_i}}$，对 F 取对数：$\ln F = \sum_i \ln \frac{1}{1-x^{v_i}}$。那么现在的问题就是求出 $\ln \frac{1}{1-x^k}$。

设 $p(x) = 1 - x^k$，$q(x) = \ln p(x)$，则

$$\frac{p'(x)}{p(x)} = q'(x)$$

$$\frac{-kx^{k-1}}{1-x^k} = q'(x)$$

$$-\sum_{i\geq 0} \frac{kx^{k+ik}}{k+ik} = q(x)$$

$$-\sum_{i\geq 1} \frac{x^{ik}}{i} = q(x)$$

LG4389 符公主的背包

考虑用生成函数表示。

已知 $\sum_{i\geq 0} x^{iv} = \frac{1}{1-x^v}$，那么要求的就是 $[x^s]\prod \frac{1}{1-x^{v_i}}$。

设 $F(x) = \prod \frac{1}{1-x^{v_i}}$，对 F 取对数：$\ln F = \sum_i \ln \frac{1}{1-x^{v_i}}$。那么现在的问题就是求出 $\ln \frac{1}{1-x^k}$。

设 $p(x) = 1 - x^k$，$q(x) = \ln p(x)$，则

$$\begin{aligned}
\frac{p'(x)}{p(x)} &= q'(x) \\
\frac{-kx^{k-1}}{1-x^k} &= q'(x) \\
-\sum_{i\geq 0} \frac{kx^{k+ik}}{k+ik} &= q(x) \\
-\sum_{i\geq 1} \frac{x^{ik}}{i} &= q(x)
\end{aligned}$$

LG4389 符公主的背包

$\ln \frac{1}{1-x^k} = -\ln(1 - x^k)$。

LG4389 符公主的背包

$\ln \frac{1}{1-x^k} = -\ln(1-x^k)$。

统计每个体积的商品个数，然后 $O(m/k)$ 给对应位置加上系数，复杂度为 $O(n \log m)$。

LG4389 符公主的背包

$\ln \frac{1}{1-x^k} = -\ln(1-x^k)$。
统计每个体积的商品个数，然后 $O(m/k)$ 给对应位置加上系数，复杂度为 $O(n\log m)$。
求和后多项式 $\exp$ 即可，总时间复杂度 $O(n\log m)$。

LG5900 无标号无根树计数

题目描述

求 n 个点的无标号无根树数量。

$1 \le n \le 2 \cdot 10^5$。

LG5900 无标号无根树计数

无根树比较麻烦，先考虑求出无标号有根树。

LG5900 无标号无根树计数

无根树比较麻烦，先考虑求出无标号有根树。
设 $F(x)$ 为无标号有根树的 OGF。

LG5900 无标号无根树计数

无根树比较麻烦，先考虑求出无标号有根树。
设 $F(x)$ 为无标号有根树的 OGF。
考虑去掉根为若干棵子树拼接而成。由于大小方案相同算同一种方案，那么可以列方程

$$F(x) = x \cdot \prod_{i \geq 1} \left(\frac{1}{1 - x^i} \right)^{f_i}$$

LG5900 无标号无根树计数

无根树比较麻烦，先考虑求出无标号有根树。
设 $F(x)$ 为无标号有根树的 OGF。
考虑去掉根为若干棵子树拼接而成。由于大小方案相同算同一种方案，那么可以列方程

$$F(x) = x \cdot \prod_{i \geq 1} \left(\frac{1}{1 - x^i} \right)^{f_i}$$

记 $G(x) = \mathcal{E}(F(x)) = \left(\frac{1}{1-x^i}\right)^{f_i}$。考虑对其取 $\ln$ 再 $\exp$，根据上一题的套路，可以得出

$$\ln G(x) = \sum_{j=1}^{\infty} \frac{1}{j} \sum_{i=1}^{\infty} f_i (x^j)^i = \sum_{j=1} \frac{F(x^j)}{j}$$

LG5900 无标号无根树计数

无根树比较麻烦，先考虑求出无标号有根树。
设 $F(x)$ 为无标号有根树的 OGF。
考虑去掉根为若干棵子树拼接而成。由于大小方案相同算同一种方案，那么可以列方程

$$F(x) = x \cdot \prod_{i\geq 1} \left(\frac{1}{1-x^i} \right)^{f_i}$$

记 $G(x) = \mathcal{E}(F(x)) = \left(\frac{1}{1-x^i}\right)^{f_i}$。考虑对其取 $\ln$ 再 $\exp$，根据上一题的套路，可以得出

$$\ln G(x) = \sum_{j=1}^{\infty} \frac{1}{j} \sum_{i=1}^{\infty} f_i (x^j)^i = \sum_{j=1} \frac{F(x^j)}{j}$$

所以 $\mathcal{E}(F(x)) = \exp\left(\sum_{j\geq 1} \frac{F(x^j)}{j}\right)$。

LG5900 无标号无根树计数

回到方程 $F(x) = x \cdot \mathcal{E}(F(x))$。这个可以直接牛顿迭代 $O(n \log n)$ 求出，但常数巨大。

LG5900 无标号无根树计数

回到方程 $F(x) = x \cdot \mathcal{E}(F(x))$。这个可以直接牛顿迭代 $O(n \log n)$ 求出，但常数巨大。

对等式两边求导后同时乘上 x 化简可以得到：

$$xF'(x) = F(x) + F(x) \sum_{n \geq 1} F'(x^n) x^n$$

LG5900 无标号无根树计数

回到方程 $F(x) = x \cdot \mathcal{E}(F(x))$。这个可以直接牛顿迭代 $O(n \log n)$ 求出，但常数巨大。
对等式两边求导后同时乘上 x 化简可以得到：

$$xF'(x) = F(x) + F(x) \sum_{n \geq 1} F'(x^n) x^n$$

设 $G(x) = \sum_n F'(x^n) x^n$，那么 $g_n = \sum_{d|n} d f_d$。
因此 $f_n = \frac{1}{n-1} \left(\sum_i f_i g_{n-i}\right)$，可以分治 NTT 解决。

LG5900 无标号无根树计数

回到方程 $F(x) = x \cdot \mathcal{E}(F(x))$。这个可以直接牛顿迭代 $O(n \log n)$ 求出，但常数巨大。
对等式两边求导后同时乘上 x 化简可以得到：

$$xF'(x) = F(x) + F(x) \sum_{n \geq 1} F'(x^n) x^n$$

设 $G(x) = \sum_n F'(x^n) x^n$，那么 $g_n = \sum_{d|n} df_d$。
因此 $f_n = \frac{1}{n-1} (\sum_i f_i g_{n-i})$，可以分治 NTT 解决。
接下来考虑计算无根树，用总方案数减去根不是重心的方案数：

$$h_n = f_n - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} f_i f_{n-i}$$

LG5900 无标号无根树计数

回到方程 $F(x) = x \cdot \mathcal{E}(F(x))$。这个可以直接牛顿迭代 $O(n \log n)$ 求出，但常数巨大。
对等式两边求导后同时乘上 x 化简可以得到：

$$xF'(x) = F(x) + F(x) \sum_{n \geq 1} F'(x^n) x^n$$

设 $G(x) = \sum_n F'(x^n) x^n$，那么 $g_n = \sum_{d|n} d f_d$。
因此 $f_n = \frac{1}{n-1} \left(\sum_i f_i g_{n-i}\right)$，可以分治 NTT 解决。
接下来考虑计算无根树，用总方案数减去根不是重心的方案数：

$$h_n = f_n - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} f_i f_{n-i}$$

但是 n 为偶数的时候重心可能有两个，重复计数当且仅当去掉两个重心之间的边两边树的结构不相同，方案数为 $\binom{f_{\frac{n}{2}}}{2}$。
总时间复杂度 $O(n \log^2 n)$。

LG5900 无标号无根树计数

Bonus：对于 $n = 1, 2, \ldots, m$，求出 n 个点的无标号无根树数量。

LG5900 无标号无根树计数

Bonus：对于 $n = 1, 2, \ldots, m$，求出 n 个点的无标号无根树数量。

$$h_n = f_n - \frac{1}{2}\left([x^n]F(x)^2 - f_{\frac{n}{2}}(f_{\frac{n}{2}} - 1)[n \bmod 2 = 0]\right)$$

求出 f 后一次卷积即可。

CF1770E Koxia and Tree

题目描述

给定一棵 n 个节点的树，树上有 k 个位置存在蝴蝶。进行 $n-1$ 轮操作：第 i 轮操作给定一条没有操作过的边 (u_i, v_i)，给其定向，有 $\frac{1}{2}$ 的概率定向为 $u_i \to v_i$，有 $\frac{1}{2}$ 的概率定向为 $v_i \to u_i$。若定向后的起点有蝴蝶且终点没有蝴蝶，则操作后起点的蝴蝶飞向终点。
求出 $n-1$ 轮操作后任取两个蝴蝶的树上距离的期望。
$2 \le k \le n \le 3 \cdot 10^5$。

CF1770E Koxia and Tree

如果蝴蝶不移动，这就是个经典题。

CF1770E Koxia and Tree

如果蝴蝶不移动，这就是个经典题。
累加每条边的贡献，贡献为这条边两边的蝴蝶数量之积。

CF1770E Koxia and Tree

如果蝴蝶不移动，这就是个经典题。
累加每条边的贡献，贡献为这条边两边的蝴蝶数量之积。
考虑蝴蝶移动造成的影响。

CF1770E Koxia and Tree

如果蝴蝶不移动，这就是个经典题。
累加每条边的贡献，贡献为这条边两边的蝴蝶数量之积。
考虑蝴蝶移动造成的影响。
设 p_u 为节点 u 上存在蝴蝶的概率，sz_u 为 u 子树内的蝴蝶数量。设当前操作的边是 (u,v)，且 u 是 v 的父亲节点。分三种情况讨论：

- 有只蝴蝶从 u 飞到 v 的概率为：$p_1 = \frac{1}{2} \times p_u \times (1-p_v)$。此时 $sz_v \leftarrow sz_v + 1$，sz_u 不变。
- 有只蝴蝶从 v 飞到 u 的概率为：$p_2 = \frac{1}{2} \times p_v \times (1-p_u)$。此时 $sz_v \leftarrow sz_v - 1$，sz_u 不变。
- 有 $1-p_1-p_2$ 的概率蝴蝶不发生移动。此时 sz_u 和 sz_v 均不发生变化。

CF1770E Koxia and Tree

如果蝴蝶不移动，这就是个经典题。
累加每条边的贡献，贡献为这条边两边的蝴蝶数量之积。
考虑蝴蝶移动造成的影响。
设 p_u 为节点 u 上存在蝴蝶的概率，sz_u 为 u 子树内的蝴蝶数量。设当前操作的边是 (u, v)，且 u 是 v 的父亲节点。分三种情况讨论：

- 有只蝴蝶从 u 飞到 v 的概率为：$p_1 = \frac{1}{2} \times p_u \times (1 - p_v)$。此时 $sz_v \leftarrow sz_v + 1$，sz_u 不变。
- 有只蝴蝶从 v 飞到 u 的概率为：$p_2 = \frac{1}{2} \times p_v \times (1 - p_u)$。此时 $sz_v \leftarrow sz_v - 1$，sz_u 不变。
- 有 $1 - p_1 - p_2$ 的概率蝴蝶不发生移动。此时 sz_u 和 sz_v 均不发生变化。

操作后，$p_u = p_v = \frac{p_u + p_v}{2}$。累加起来为所有选取方案的贡献和，期望也就是乘上 $\frac{2}{k(k-1)}$。总时间复杂度 $O(n)$。

CF1615F LEGOndary Grandmaster

题目描述

给定两个长度一样的 01 串 s,t，其中有些位置上的字符已经忘记了，忘记的位置上是 ?。
定义 $f(s,t)$ 表示最少使得 $s=t$ 的操作数。操作是选择 s 中相邻的两个一样的字符，同时进行反转，即 00 变 11，11 变 00。
计算出对于所有将 s,t 中的 ? 填成 0 或 1 的方案，令 s 填完后的字符串为 s'，t 填完后的字符串为 t'，所有 $f(s',t')$ 的总和。
$2\le|s|\le 2000$。

CF1615F LEGOndary Grandmaster

考虑对于填完的两个字符串 s, t，计算使得 $s = t$ 的最小操作数。

CF1615F LEGOndary Grandmaster

考虑对于填完的两个字符串 s, t，计算使得 $s = t$ 的最小操作数。

Trick

由于一次操作恰好反转一个奇数位上的字符和一个偶数位上的字符，那么考虑将所有的奇数位上的字符反转。此时反转原序列中的两个相邻的相同的字符，相当于在现序列上交换这两个字符；原序列中的两个不相邻的字符在现序列中是两个一样的字符，交换也不会影响。所以可以把原序列的操作转换为现序列的交换两个相邻字符。

CF1615F LEGOndary Grandmaster

考虑对于填完的两个字符串 s, t，计算使得 $s = t$ 的最小操作数。

Trick

由于一次操作恰好反转一个奇数位上的字符和一个偶数位上的字符，那么考虑将所有的奇数位上的字符反转。此时反转原序列中的两个相邻的相同的字符，相当于在现序列上交换这两个字符；原序列中的两个不相邻的字符在现序列中是两个一样的字符，交换也不会影响。所以可以把原序列的操作转换为现序列的交换两个相邻字符。

那么现在若对 s, t 都进行此变换，得到的结果字符串为 s', t'。记 s' 中的 1 的位置是 $p_1 < p_2 < \cdots < p_k$，t' 中的 1 的位置是 $q_1 < q_2 < \cdots < q_k$，则最小操作数为 $\sum_{i=1}^{k} |p_i - q_i|$。可以直接设 dp 解决，但其实还可以进一步化简。

CF1615F LEGOndary Grandmaster

考虑将 $|p_i - q_i|$ 分到 $[\min(p_i, q_i), \max(p_i, q_i))$ 上，那么此时形成的序列即是 s' 的前缀 1 数量构成的序列与 t' 的前缀 1 数量构成的序列的差值。也就是记 a_i 为 $s'[1...i]$ 中 1 的数量，b_i 为 $t'[1...i]$ 中 1 的数量，答案就是 $\sum_{i=1}^{n} |a_i - b_i|$，而 $|a_i - b_i|$ 每次的变化量 $\Delta \in \{-1, 0, 1\}$，所以直接 dp 即可。

CF1615F LEGOndary Grandmaster

考虑将 $|p_i - q_i|$ 分到 $[\min(p_i, q_i), \max(p_i, q_i))$ 上，那么此时形成的序列即是 s' 的前缀 1 数量构成的序列与 t' 的前缀 1 数量构成的序列的差值。也就是记 a_i 为 $s'[1...i]$ 中 1 的数量，b_i 为 $t'[1...i]$ 中 1 的数量，答案就是 $\sum\limits_{i=1}^{n} |a_i - b_i|$，而 $|a_i - b_i|$ 每次的变化量 $\Delta \in \{-1, 0, 1\}$，所以直接 dp 即可。

设 $pre_{i,j}$ 表示确定了 s', t' 中的 $[1...i]$ 位置，$\sum\limits_{k=1}^{i}(a_k - b_k)$ 的值为 j 的方案数，$suf_{i,j}$ 表示确定了 $[i...n]$ 位置，$\sum\limits_{k=i}^{n}(a_k - b_k)$ 的值为 j 的方案数。

则答案为 $\sum\limits_{i=1}^{n}\sum\limits_{\Delta=-i}^{i} |\Delta| \cdot pre_{i,\Delta} \cdot suf_{i+1,-\Delta}$。

总时间复杂度 $O(n^2)$。

[ZJOI2022] 树

题目描述

九条可怜要生成两棵 n 个节点的树。第一棵树的生成方式如下：

- 节点 1 为树的根。
- 对于节点 $i \in (1, n]$，从 $[1, i)$ 中随机选择一个节点作为父亲节点。

第二棵树的生成方式如下：

- 节点 n 为树的根。
- 对于节点 $i \in [1, n)$，从 $(i, n]$ 中随机选择一个节点作为父亲节点。

要求生成的两棵树的方案数，满足第一棵树中的叶子节点都是第二棵树中的非叶子节点，第一棵树中的非叶子节点都是第二棵树中的叶子节点。求出答案对 M 取模的结果。

$2 \leq n \leq 500, 10 \leq M \leq 2^{30}$。

[ZJOI2022] 树

设 $f(S)$ 表示第一棵树的非叶子节点集合**恰好**是 S 的方案数，设 $g(T)$ 表示第二棵树的非叶子节点集合**恰好**是 T 的方案数。则答案为

$$\text{Ans} = \sum_{S\cap T=\emptyset, S\cup T=\{1,2,\dots,n\}} f(S)g(T)$$

[ZJOI2022] 树

设 $f(S)$ 表示第一棵树的非叶子节点集合**恰好**是 S 的方案数，设 $g(T)$ 表示第二棵树的非叶子节点集合**恰好**是 T 的方案数。则答案为

$$\text{Ans} = \sum_{S\cap T=\emptyset, S\cup T=\{1,2,\dots,n\}} f(S)g(T)$$

考虑容斥掉“恰好”这一限制，设 $f'(S)$ 表示第一棵树的非叶子节点集合为 S 的子集的方案数，设 $g'(T)$ 表示第二棵树的非叶子节点集合为 T 的子集的方案数。则有

$$\begin{aligned}\text{Ans} &= \sum_{S\cap T=\emptyset, S\cup T=\{1,2,\dots,n\}} \sum_{S'\subseteq S, T'\subseteq T} f'(S')g'(T')(-1)^{|S|+|T|-|S'|-|T'|} \\ &= \sum_{S'\cap T'=\emptyset} f'(S')g'(T')(-2)^{n-|S'|-|T'|}\end{aligned}$$

[ZJOI2022] 树

设 $f(S)$ 表示第一棵树的非叶子节点集合**恰好**是 S 的方案数，设 $g(T)$ 表示第二棵树的非叶子节点集合**恰好**是 T 的方案数。则答案为

$$\text{Ans} = \sum_{S\cap T=\emptyset, S\cup T=\{1,2,\dots,n\}} f(S)g(T)$$

考虑容斥掉“恰好”这一限制，设 $f'(S)$ 表示第一棵树的非叶子节点集合为 S 的子集的方案数，设 $g'(T)$ 表示第二棵树的非叶子节点集合为 T 的子集的方案数。则有

$$\begin{aligned}\text{Ans} &= \sum_{S\cap T=\emptyset, S\cup T=\{1,2,\dots,n\}} \sum_{S'\subseteq S, T'\subseteq T} f'(S')g'(T')(-1)^{|S|+|T|-|S'|-|T'|} \\ &= \sum_{S'\cap T'=\emptyset} f'(S')g'(T')(-2)^{n-|S'|-|T'|}\end{aligned}$$

[ZJOI2022] 树

考虑 DP。

[ZJOI2022] 树

考虑 DP。
设 $dp_{i,j,k}$ 表示考虑到节点 i，$|\{1,2,\ldots,i\}\cap S'|=j|$，$\{i+1,i+2,\ldots,n\}\cap T'|=k$ 的方案数。

[ZJOI2022] 树

考虑 DP。
设 $dp_{i,j,k}$ 表示考虑到节点 i，$|\{1,2,\ldots,i\}\cap S'|=j|$，$\{i+1,i+2,\ldots,n\}\cap T'|=k$ 的方案数。
转移方程考虑节点 i 的情况：

- $i\in S'$，$dp_{i,j+1,k}\leftarrow dp_{i-1,j,k}\times j\times k$。
- $i\in T'$，$dp_{i,j,k-1}\leftarrow dp_{i-1,j,k}\times j\times k$。
- $i\notin S',i\notin T'$，$dp_{i,j,k}\leftarrow dp_{i-1,j,k}\times(-2)\times j\times k$。

$j\times k$ 即考虑选择节点 i 在两棵树中的父亲节点的方案数。
总时间复杂度 $O(n^3)$。

SOJ1718 丝线

题目描述

有 $n = 2A + B$ 根线，其中

- 有 A 根线，两端都是红色。
- 有 A 根线，两端都是绿色。
- 有 B 根线，一端红色一端绿色。

因此恰好有 n 个端点时红色，n 个端点是绿色。
现在把红色端点和绿色端点从 $n!$ 种方案中随机选取一种匹配，并且把匹配的端点连在一起。求出期望会得到多少个环。
$1 \le A, B \le 10^6$。

SOJ1718 丝线

红绿线是一种很有趣味的线。任何一个绿端点接到它的红端点之后，剩下的仍然是一个绿端点，相当于无事发生，只是少了一条红绿线而已。

SOJ1718 丝线

红绿线是一种很有趣味的线。任何一个绿端点接到它的红端点之后，剩下的仍然是一个绿端点，相当于无事发生，只是少了一条红绿线而已。
所以如果设 $f(A, B)$ 表示答案，那么 $f(A, B)$ 几乎就是 $f(A, B-1)$。
为什么是几乎呢？因为有可能是自己接到自己了，此时就发生了一点事情。
因此 $f(A, B) = f(A, B-1) + \frac{1}{2A+B}$。

SOJ1718 丝线

红绿线是一种很有趣味的线。任何一个绿端点接到它的红端点之后，剩下的仍然是一个绿端点，相当于无事发生，只是少了一条红绿线而已。
所以如果设 $f(A,B)$ 表示答案，那么 $f(A,B)$ 几乎就是 $f(A,B-1)$。
为什么是几乎呢？因为有可能是自己接到自己了，此时就发生了一点事情。
因此 $f(A,B)=f(A,B-1)+\frac{1}{2A+B}$。
如果 $B=0$ 就随便选两条线接起来就行了，反正也没有任何区别，两条线连起来后等价于一条红绿线：$f(A,0)=f(A-1,1)+1$。
时间复杂度 $O(A+B)$。

SOJ1720 绘画

题目描述

有一个 $H \times W$ 的网格，每个格子初始都是白色。
可以进行若干次操作，每次可以把一列染黑，或者把一个从左上到右下的对角线染黑。
求出最终能得到多少种不同的网格。
$1 \le H, W \le 50$。

SOJ1720 绘画

先复制粘贴一个复杂度较高的**标算**做法，$O(W^3(H+W))$。
考虑把一个结果映射到唯一一个操作集合上。我选择优先操作对角线：只要一个对角线全是黑的，就认为这条对角线被选了。

SOJ1720 绘画

先复制粘贴一个复杂度较高的**标算**做法，$O(W^3(H+W))$。
考虑把一个结果映射到唯一一个操作集合上。我选择优先操作对角线：只要一个对角线全是黑的，就认为这条对角线被选了。
确定了对角线的结果之后，每一列有三种情况：

- 这一列上的对角线全都被操作了，所以它是否操作不影响结果。称这一列被标满了。
- 对角线没有全都被操作，而这一列也不选。
- 对角线没有全都被操作，但这一列要选。

SOJ1720 绘画

先复制粘贴一个复杂度较高的**标算**做法，$O(W^3(H+W))$。
考虑把一个结果映射到唯一一个操作集合上。我选择优先操作对角线：只要一个对角线全是黑的，就认为这条对角线被选了。
确定了对角线的结果之后，每一列有三种情况：

- 这一列上的对角线全都被操作了，所以它是否操作不影响结果。称这一列被标满了。
- 对角线没有全都被操作，而这一列也不选。
- 对角线没有全都被操作，但这一列要选。

现在我们就可以用几个条件来确定一个操作集合是否合法：

- 未选对角线在前 n 列里存在一个未选的列。
- 未选对角线在前 n 列里不存在一个标满的列。
- 未标满的列在后 n 个对角线里存在一个未选的对角线。

所以直接设 $dp_{i,j,k,l}$ 表示考虑到第 i 列/对角线，上一个未选列在 j ，最早的未标满且未搞定的列在 k，上一个标满的列在 l 的方案数，然后硬转移即可。

SOJ1720 绘画

然而还有一种复杂度 $O(W^2(H+W))$ 的做法。
先将对角线从左下到右上标号为 $1, 2, \ldots, H+W-1$。

SOJ1720 绘画

然而还有一种复杂度 $O(W^2(H+W))$ 的做法。
先将对角线从左下到右上标号为 $1,2,\ldots,H+W-1$。
考虑先覆盖列，再覆盖对角线。从左往右扫，在已经确定了前 $i-1$ 列且第 i 列没有覆盖的情况下，考虑 $1\sim i+H-1$ 条对角线。假设上一个没被覆盖的列为 j，那么从 $\max(j+H,i)$ 到 $i+H-1$ 的对角线选不选都是不同的方案。

SOJ1720 绘画

然而还有一种复杂度 $O(W^2(H+W))$ 的做法。
先将对角线从左下到右上标号为 $1,2,\dots,H+W-1$。
考虑先覆盖列，再覆盖对角线。从左往右扫，在已经确定了前 $i-1$ 列且第 i 列没有覆盖的情况下，考虑 $1\sim i+H-1$ 条对角线。假设上一个没被覆盖的列为 j，那么从 $\max(j+H,i)$ 到 $i+H-1$ 的对角线选不选都是不同的方案。
因此记 $dp_{i,k}$ 表示确定了前 $i-1$ 列的状态且第 i 列不被覆盖，最后一个覆盖的对角线是 k，考虑对角线 $1\sim i+H-1$ 的方案数。转移时枚举上一个没被覆盖的列 j 即可。

Coloring the ring

题目描述

给定 r 个红球、g 个绿球和 b 个蓝球，每个球都有一个独特的编号。把这些球排成一个长度为 $r+g+b$ 的环，满足以下条件：

- 对于任意两个红球，它们之间的两段上都有至少 p 个绿球；
- 对于任意两个绿球，它们之间的两段上都有至少 q 个蓝球。

求方案数。

$0 \le r, g, b, p, q \le 10^6, r, g \ne 1, rp \le g, gq \le b, r+g+b > 0$。

Coloring the ring

先讲一种不那么自然的做法。

Coloring the ring

先讲一种不那么自然的做法。
简单的观察一下，题目中的条件等价于：对于 r 个环上的红球，相邻两个红球之间至少有 p 个绿球；对于 g 个绿球，相邻两个绿球之间至少有 b 个蓝球。

Coloring the ring

先讲一种不那么自然的做法。
简单的观察一下，题目中的条件等价于：对于 r 个环上的红球，相邻两个红球之间至少有 p 个绿球；对于 g 个绿球，相邻两个绿球之间至少有 b 个蓝球。
考虑先把环断开，$p=0$ 的时候钦定第一个是绿球，然后相当于 b 个球分成 g 段，每段 $\geq q$ 个。这个插板法即可。

Coloring the ring

先讲一种不那么自然的做法。
简单的观察一下，题目中的条件等价于：对于 r 个环上的红球，相邻两个红球之间至少有 p 个绿球；对于 g 个绿球，相邻两个绿球之间至少有 b 个蓝球。
考虑先把环断开，$p=0$ 的时候钦定第一个是绿球，然后相当于 b 个球分成 g 段，每段 $\geq q$ 个。这个插板法即可。
$p>0$ 时考虑先放红球和绿球。这也是一个插板法。然后加入蓝球，蓝球会被红球分开，考虑枚举左边有 $\geq q$ 个蓝球的红球个数 i，那就相当于将 b 个球放到 $g+i$ 段里，其中有 g 个段满足至少有 q 个球，方案数为 $\binom{b-qg+g+i-1}{g+i-1}$，剩下 $r-i$ 个红球的左边都 $<q$，方案数为 q^{r-i}。
时间复杂度 $O(r)$。

Coloring the ring

下面介绍另一种用生成函数推的做法。

Coloring the ring

下面介绍另一种用生成函数推的做法。
首先还是考虑先放完红球和绿球，然后加入蓝球。设第 i 个绿球和第 $i \bmod g + 1$ 个绿球之间有 l_i 个蓝球。则相邻绿球之间既有红球又有蓝球会对答案产生贡献，为 $\prod(l_i + 1)$。

Coloring the ring

下面介绍另一种用生成函数推的做法。
首先还是考虑先放完红球和绿球，然后加入蓝球。设第 i 个绿球和第 $i \bmod g + 1$ 个绿球之间有 l_i 个蓝球。则相邻绿球之间既有红球又有蓝球会对答案产生贡献，为 $\prod(l_i + 1)$。
对所有的 l_i 减去 t，考虑枚举对答案产生贡献的 l_i 的和 t，那么不对答案产生贡献的 l_i 可以用插板法解决，即 $\binom{b-gq-t+g-r-1}{g-r-1}$。

Coloring the ring

下面介绍另一种用生成函数推的做法。
首先还是考虑先放完红球和绿球，然后加入蓝球。设第 i 个绿球和第 $i \bmod g+1$ 个绿球之间有 l_i 个蓝球。则相邻绿球之间既有红球又有蓝球会对答案产生贡献，为 $\prod(l_i+1)$。
对所有的 l_i 减去 t，考虑枚举对答案产生贡献的 l_i 的和 t，那么不对答案产生贡献的 l_i 可以用插板法解决，即 $\binom{b-gq-t+g-r-1}{g-r-1}$。
那么现在就是要算和为 t 的 r 个 l_i 的贡献，即
$\sum_l \prod_{i=1}^r (l_i+q+1) \quad (l_1+l_2+\ldots+l_r=t;\ l_i \geq 0)$

Coloring the ring

下面介绍另一种用生成函数推的做法。
首先还是考虑先放完红球和绿球，然后加入蓝球。设第 i 个绿球和第 $i \bmod g+1$ 个绿球之间有 l_i 个蓝球。则相邻绿球之间既有红球又有蓝球会对答案产生贡献，为 $\prod(l_i+1)$。
对所有的 l_i 减去 t，考虑枚举对答案产生贡献的 l_i 的和 t，那么不对答案产生贡献的 l_i 可以用插板法解决，即 $\binom{b-gq-t+g-r-1}{g-r-1}$。
那么现在就是要算和为 t 的 r 个 l_i 的贡献，即
$\sum_l \prod_{i=1}^r (l_i+q+1) \quad (l_1+l_2+\ldots+l_r=t;\ l_i \geq 0)$ 设
$F(x)=(q+1)x^0+(q+2)x^1+\cdots+(q+n+1)x^n+\cdots=\frac{-qx+q+1}{(1-x)^2}$ 答案为 $\sum_t \binom{b-gq-t+g-r-1}{g-r-1}[x^t](F(x)^r)$。直接算是平方级别的，注意到实际上就是 $[x^{b-gq}]\frac{1}{(1-x)^{g-r}}\frac{(-qx+q+1)^r}{(1-x)^{2r}}$。那么分子二项式展开，分母非常经典，最后求 $b-gq$ 处的值即可。

Coloring the ring

下面介绍另一种用生成函数推的做法。
首先还是考虑先放完红球和绿球，然后加入蓝球。设第 i 个绿球和第 $i \bmod g+1$ 个绿球之间有 l_i 个蓝球。则相邻绿球之间既有红球又有蓝球会对答案产生贡献，为 $\prod(l_i+1)$。
对所有的 l_i 减去 t，考虑枚举对答案产生贡献的 l_i 的和 t，那么不对答案产生贡献的 l_i 可以用插板法解决，即 $\binom{b-gq-t+g-r-1}{g-r-1}$。
那么现在就是要算和为 t 的 r 个 l_i 的贡献，即
$\sum_l \prod_{i=1}^r (l_i+q+1) \quad (l_1+l_2+\ldots+l_r=t;\ l_i \ge 0)$ 设
$F(x)=(q+1)x^0+(q+2)x^1+\cdots+(q+n+1)x^n+\cdots=\frac{-qx+q+1}{(1-x)^2}$ 答案为 $\sum_t \binom{b-gq-t+g-r-1}{g-r-1}[x^t](F(x)^r)$。直接算是平方级别的，注意到实际上就是 $[x^{b-gq}]\frac{1}{(1-x)^{g-r}}\frac{(-qx+q+1)^r}{(1-x)^{2r}}$。那么分子二项式展开，分母非常经典，最后求 $b-gq$ 处的值即可。
时间复杂度 $O(r)$。

Walking on the Table

题目描述

给定一个 $n \times m$ 的方格矩阵，小 Z 要从 $(1,1)$ 走到 (n,m)，得分即是途经的数的和。行走时遵循以下规则：

- 1. 到达了下边界，则往右走.
- 2. 到达了右边界，则往下走.
- 3. 下面和右边都有数. 如果右边的数 $>=$ 下面的数，往右走；否则往下走.

现在可以在每个位置上填上一个取值范围在 $[0, S]$ 中的整数，满足 $(1,1)$ 位置上的数是 0，问有多少种填充方案使得小 Z 按照规则走到终点时的得分 **恰好** 为 S。

$1 \leq n, m, S \leq 10^6$。

Walking on the Table

因为贴着边界走是特殊的情况，所以先考虑有多少步是特殊情况。考虑最后贴着下边界走的情况，右边界同理。

Walking on the Table

因为贴着边界走是特殊的情况，所以先考虑有多少步是特殊情况。考虑最后贴着下边界走的情况，右边界同理。
枚举最后 x 格是贴着下边界走的（贴着下边界走了 $x-1$ 步），那么正常情况下走的步数中，$n-1$ 步是向下走的，$m-x$ 步是向右走的。受路径影响的格子有 $2(n-1+m-x)+x=2n+2m-x-2$ 个（即会被路径比较、选择到的格子）。剩下的 $nm-2n-2m+x+2$ 个格子可以任意填充 $[0,S]$ 中的数，则不受路径影响的填充方案数为：$(S+1)^{nm-2n-2m+x+2}$。

Walking on the Table

因为贴着边界走是特殊的情况，所以先考虑有多少步是特殊情况。考虑最后贴着下边界走的情况，右边界同理。
枚举最后 x 格是贴着下边界走的（贴着下边界走了 $x-1$ 步），那么正常情况下走的步数中，$n-1$ 步是向下走的，$m-x$ 步是向右走的。受路径影响的格子有 $2(n-1+m-x)+x=2n+2m-x-2$ 个（即会被路径比较、选择到的格子）。剩下的 $nm-2n-2m+x+2$ 个格子可以任意填充 $[0,S]$ 中的数，则不受路径影响的填充方案数为：
$(S+1)^{nm-2n-2m+x+2}$。
因为不同路径对应的矩阵也不同，所以还要计算有多少种不同的路径，注意最后一步到达下边界时必定是向下走的，不然贴底的格子就不止 x 个，选择路径的方案数为 $\binom{n-2+m-x}{n-2}$。

Walking on the Table

因为贴着边界走是特殊的情况，所以先考虑有多少步是特殊情况。考虑最后贴着下边界走的情况，右边界同理。
枚举最后 x 格是贴着下边界走的（贴着下边界走了 $x-1$ 步），那么正常情况下走的步数中，$n-1$ 步是向下走的，$m-x$ 步是向右走的。受路径影响的格子有 $2(n-1+m-x)+x=2n+2m-x-2$ 个（即会被路径比较、选择到的格子）。剩下的 $nm-2n-2m+x+2$ 个格子可以任意填充 $[0,S]$ 中的数，则不受路径影响的填充方案数为：
$(S+1)^{nm-2n-2m+x+2}$。
因为不同路径对应的矩阵也不同，所以还要计算有多少种不同的路径，注意最后一步到达下边界时必定是向下走的，不然贴底的格子就不止 x 个，选择路径的方案数为 $\binom{n-2+m-x}{n-2}$。

Walking on the Table

接下来考虑对路径有影响的那些格子，需要给路径上每一步就赋上一个权值，使得权值和恰好等于 S。注意在前面 $n-1+m-x$ 步中，每走一步都要比较一下，所以如果往下走时，那么下方的格子 **严格大于** 右方的格子；往右走时，那么右方的格子 **大于等于** 下方的格子。若钦定往下走且下方的格子为 w 时，右方的格子有 $[0, w-1]$ 共 w 种方案；若钦定往右走且右方的格子为 w 时，下方的格子有 $[0, w]$ 共 $w+1$ 种方案。

Walking on the Table

接下来考虑对路径有影响的那些格子，需要给路径上每一步就赋上一个权值，使得权值和恰好等于 S。注意在前面 $n-1+m-x$ 步中，每走一步都要比较一下，所以如果往下走时，那么下方的格子 **严格大于** 右方的格子；往右走时，那么右方的格子 **大于等于** 下方的格子。若钦定往下走且下方的格子为 w 时，右方的格子有 $[0,w-1]$ 共 w 种方案；若钦定往右走且右方的格子为 w 时，下方的格子有 $[0,w]$ 共 $w+1$ 种方案。形式化地，这部分的方案数的表达式为（令 w_i 为第 i 步的权值）：

$\sum_{w_1=0}^{S}\sum_{w_2=0}^{S}\cdots\sum_{w_{n+m-2}=0}^{S} w_1w_2\cdots w_{n-1}(w_n+1)$
$(w_{n+1}+1)\cdots(w_{n-1+m-x}+1)\cdot[\sum_{i=1}^{n+m-2} w_i = S]$

Walking on the Table

式子很繁琐，考虑化简它，有一个较为简单的公式：

Formula

$$\sum_{x=0}^{S}\sum_{y=0}^{S} xy \cdot [x+y=S] = \binom{S+1}{3}$$

考虑这个式子的组合意义，从排成一行的 $S+1$ 个点中选取一个作为中间点，再从两边（左边 x 个，右边 y 个，$x+y=S$）分别选出一个点的方案数等于从 $S+1$ 的点中选出 3 个点的方案数。

Walking on the Table

式子很繁琐，考虑化简它，有一个较为简单的公式：

Formula

$$\sum_{x=0}^{S}\sum_{y=0}^{S} xy \cdot [x+y=S] = \binom{S+1}{3}$$

考虑这个式子的组合意义，从排成一行的 $S+1$ 个点中选取一个作为中间点，再从两边（左边 x 个，右边 y 个，$x+y=S$）分别选出一个点的方案数等于从 $S+1$ 的点中选出 3 个点的方案数。
推广一下易知刚才的式子等于 $\binom{S+n+2m-x-3}{2n+2m-x-4}$。
所以最后 x 格贴着下边界的方案数就是：

$$(S+1)^{nm-2n-2m+x+2}\binom{n-2+m-x}{n-2}\binom{S+n+2m-x-3}{2n+2m-x-4}$$

LG8768 [蓝桥杯 2021 国 A] 积木

题目描述

小蓝要搭 n 行积木，第一行摆了 $H_1 = w$ 块积木。从第二行开始，第 i 行的积木数量都至少比上一行多 L，至多比上一行多 R (当 $L = 0$ 时表示可以和上一行的积木数量相同)，即

$$H_{i-1} + L \leq H_i \leq H_{i-1} + R$$

给定 x, y, z，求满足上述条件的方案中，有多少种方案满足 $H_y = zH_x$。

$1 \leq n \leq 5 \cdot 10^5, 1 \leq w \leq 10^9, 0 \leq L \leq R \leq 40, 1 \leq x < y \leq n, 0 \leq z \leq 10^9$。

LG8768 [蓝桥杯 2021 国 A] 积木

首先注意到 $y+1\sim n$ 行的积木不会对答案造成影响，方案数为 $(R-L+1)^{n-y}$。

LG8768 [蓝桥杯 2021 国 A] 积木

首先注意到 $y+1\sim n$ 行的积木不会对答案造成影响，方案数为 $(R-L+1)^{n-y}$。
假设 $H_x-H_1=q$，那么 $H_y-H_x=(z-1)(w+q)$。

LG8768 [蓝桥杯 2021 国 A] 积木

首先注意到 $y+1\sim n$ 行的积木不会对答案造成影响，方案数为 $(R-L+1)^{n-y}$。
假设 $H_x-H_1=q$，那么 $H_y-H_x=(z-1)(w+q)$。
那么现在问题转化为求 $H_r-H_l=i$ 的方案数。设 $n=r-l$，F 为答案的 OGF，则有

$$F=(x^L+x^{L+1}+\cdots+x^R)^n=\left(\frac{x^L-x^{R+1}}{1-x}\right)^n=x^{Ln}\left(\frac{1-x^{R-L+1}}{1-x}\right)^n$$

LG8768 [蓝桥杯 2021 国 A] 积木

首先注意到 $y+1\sim n$ 行的积木不会对答案造成影响，方案数为 $(R-L+1)^{n-y}$。
假设 $H_x-H_1=q$，那么 $H_y-H_x=(z-1)(w+q)$。
那么现在问题转化为求 $H_r-H_l=i$ 的方案数。设 $n=r-l$，F 为答案的 OGF，则有

$$F=(x^L+x^{L+1}+\cdots+x^R)^n=\left(\frac{x^L-x^{R+1}}{1-x}\right)^n=x^{Ln}\left(\frac{1-x^{R-L+1}}{1-x}\right)^n$$

令 $k=R-L+1$。要求的即为 $F=\left(\frac{1-x^k}{1-x}\right)^n$ 的各项系数。

LG8768 [蓝桥杯 2021 国 A] 积木

首先注意到 $y+1 \sim n$ 行的积木不会对答案造成影响，方案数为 $(R-L+1)^{n-y}$。
假设 $H_x - H_1 = q$，那么 $H_y - H_x = (z-1)(w+q)$。
那么现在问题转化为求 $H_r - H_l = i$ 的方案数。设 $n = r - l$，F 为答案的 OGF，则有

$$F = (x^L + x^{L+1} + \cdots + x^R)^n = \left(\frac{x^L - x^{R+1}}{1-x}\right)^n = x^{Ln}\left(\frac{1-x^{R-L+1}}{1-x}\right)^n$$

令 $k = R - L + 1$。要求的即为 $F = \left(\frac{1-x^k}{1-x}\right)^n$ 的各项系数。
对 F 求导，

$$\begin{aligned} F' &= n \cdot \left(\frac{1-x^k}{1-x}\right)^{n-1} \cdot \frac{(k-1)x^k - kx^{k-1} + 1}{(1-x)^2} \\ &= F \cdot \frac{n \cdot ((k-1)x^k - kx^{k-1} + 1)}{(1-x)(1-x^k)} \end{aligned}$$

LG8768 [蓝桥杯 2021 国 A] 积木

现在有 $(1-x)(1-x^k)F' = F \cdot n \cdot ((k-1)x^k - kx^{k-1} + 1)$。

LG8768 [蓝桥杯 2021 国 A] 积木

现在有 $(1-x)(1-x^k)F' = F \cdot n \cdot ((k-1)x^k - kx^{k-1} + 1)$。
左右两边提取 $[x^m]$ 即可得到 F 的整式递推：

$$\begin{aligned}&(m-k)f_{m-k} - (m-k+1)f_{m-k+1} - mf_m + (m+1)f_{m+1}\\=&n(k-1)f_{m-k} - nkf_{m-k+1} + nf_m\end{aligned}$$

LG8768 [蓝桥杯 2021 国 A] 积木

现在有 $(1-x)(1-x^k)F' = F \cdot n \cdot ((k-1)x^k - kx^{k-1} + 1)$。
左右两边提取 $[x^m]$ 即可得到 F 的整式递推：

$$\begin{aligned}&(m-k)f_{m-k} - (m-k+1)f_{m-k+1} - mf_m + (m+1)f_{m+1}\\=&n(k-1)f_{m-k} - nkf_{m-k+1} + nf_m\end{aligned}$$

直接 $O(n(R-L))$ 递推即可。

SOJ1712 你不过 t2 你过啥题啊

题目描述

给定一棵 n 个点的 k 叉树。根节点是 1，非叶子节点的子节点有序。
定义两棵 k 叉树相同当且仅当：

- 两棵树都是空的。
- 两棵树的子节点按照顺序分别对应相同。即对于每个 $i(1 \le i \le k)$，第一棵树以第 i 个子节点为根的子树与第二棵树以第 i 个子节点为根的子树相同。

如果两棵 k 叉树不同，定义两棵 k 叉树的好坏关系为：

- 如果两棵 k 叉树的大小不同，则更大的 k 叉树更好。
- 如果两棵 k 叉树的大小相同，则依次按顺序比较两棵 k 叉树以每个子节点为根的子树的好坏关系，最先比较出更好的关系的子树所在的 k 叉树更好。

现在你要求出所有非空 k 叉树中比给定的 k 叉树坏的树的数量。
$1 \le n \le 10^5, 1 \le k \le 10$。

SOJ1712 你不过 t2 你过啥题啊

Hint：考虑 $k = 2$ 时怎么做。

SOJ1712 你不过 t2 你过啥题啊

Hint：考虑 $k=2$ 时怎么做。
以下是 $k=2$ 时的情况。
设节点 x 的左右节点分别为 $lson, rson$。$size_x$ 表示以 x 为根的子树的大小。
设 $f(x)$ 表示大小与 x 相同但比以 x 为根的子树更坏的二叉树的数量。则

$$f(x) = f(lson)H(size_{rson}) + f(rson) + \sum_{i=0}^{size_{lson}-1} H(i)H(size_x - 1 - i)。$$

其中 $H(n)$ 是卡特兰数的第 n 项。直接 dp 复杂度是 $O(n^2)$。

SOJ1712 你不过 t2 你过啥题啊

Hint：考虑 $k=2$ 时怎么做。
以下是 $k=2$ 时的情况。
设节点 x 的左右节点分别为 $lson, rson$。$size_x$ 表示以 x 为根的子树的大小。
设 $f(x)$ 表示大小与 x 相同但比以 x 为根的子树更坏的二叉树的数量。
则

$$f(x) = f(lson)H(size_{rson}) + f(rson) + \sum_{i=0}^{size_{lson}-1} H(i)H(size_x - 1 - i)。$$

其中 $H(n)$ 是卡特兰数的第 n 项。直接 dp 复杂度是 $O(n^2)$。
考虑优化，算法瓶颈在于后面的求和式子。
首先我们知道 $H(n) = \sum_{i=0}^{n-1} H(i)H(n-1-i)$。
而求和上标为 $size_{lson}-1$，那么考虑 $size_{lson} \le \frac{size_x}{2}$ 时按照原式算，
而 $size_{lson} > \frac{size_x}{2}$ 的时候用 $H(size_x) - \sum_{i=size_{lson}}^{size_x-1} H(i)H(size_x-1-i)$ 计算。时间复杂度降为 $O(n\log n)$。

SOJ1712 你不过 t2 你过啥题啊

接下来考虑 $k \leq 10$。

SOJ1712 你不过 t2 你过啥题啊

接下来考虑 $k \le 10$。
可以参考上面的思路，考虑 n 个点的 k 叉树有 $F(n,k)$ 棵。

$$F(n,k) = \frac{\binom{kn}{n}}{(k-1)n+1}$$

SOJ1712 你不过 t2 你过啥题啊

接下来考虑 $k \leq 10$。
可以参考上面的思路，考虑 n 个点的 k 叉树有 $F(n,k)$ 棵。

$$F(n,k) = \frac{\binom{kn}{n}}{(k-1)n+1}$$

那么我们可以依次枚举第 i 棵子树，分两种情况进行讨论。

- 前 $i-1$ 棵子树都一模一样，第 i 棵子树大小一样但更差。
- 前 $i-1$ 棵子树都一模一样，第 i 棵子树大小更小更差。

SOJ1712 你不过 t2 你过啥题啊

接下来考虑 $k \le 10$。
可以参考上面的思路，考虑 n 个点的 k 叉树有 $F(n,k)$ 棵。

$$F(n,k) = \frac{\binom{kn}{n}}{(k-1)n+1}$$

那么我们可以依次枚举第 i 棵子树，分两种情况进行讨论。

- 前 $i-1$ 棵子树都一模一样，第 i 棵子树大小一样但更差。
- 前 $i-1$ 棵子树都一模一样，第 i 棵子树大小更小更差。

第一种情况的答案就是

$$\sum_{i=1}^{k} f(son_i)\Big(\sum_{p_{i+1}+p_{i+2}+\ldots+p_k=\sum_{j=i+1}^{n} size_{son_j}} \prod_{j=i+1}^{n} F(p_j,k)\Big)$$

发现后面一段就是要求 $[x^n]F^m$，设 $G(n,m,k)$ 表示将 n 个点分配到 m 棵 k 叉树里形成的不同 k 叉树组的方案数。也即 $[x^n]F^m$。

SOJ1712 你不过 t2 你过啥题啊

$$G(n,m,k) = \binom{kn+m-1}{n} \cdot \frac{m}{(k-1)n+m}$$

那么就可以快速求得第一种情况的答案。

SOJ1712 你不过 t2 你过啥题啊

$$G(n,m,k)=\binom{kn+m-1}{n}\cdot\frac{m}{(k-1)n+m}$$

那么就可以快速求得第一种情况的答案。
第二种情况的答案就是

$$\sum_{i=1}^{k}\sum_{j=0}^{size_{son_i}-1}F(j,k)\cdot(\sum_{p_{i+1}+p_{i+2}+\ldots+p_k=(\sum_{l=i+1}^{n}size_{son_l})+size_{son_i}-j}\prod_{l=i+1}^{n}F(p_l,k))$$

同理，可以把括号内的化为 G 函数。

$$\sum_{i=1}^{k}\sum_{j=0}^{size_{son_i}-1}F(j,k)\cdot G((\sum_{l=i+1}^{n}size_{son_l})+size_{son_i}-j,k-i-1,k)$$

SOJ1712 你不过 t2 你过啥题啊

$$G(n,m,k)=\binom{kn+m-1}{n}\cdot\frac{m}{(k-1)n+m}$$

那么就可以快速求得第一种情况的答案。
第二种情况的答案就是

$$\sum_{i=1}^{k}\sum_{j=0}^{size_{son_i}-1}F(j,k)\cdot(\sum_{p_{i+1}+p_{i+2}+\ldots+p_k=(\sum_{l=i+1}^{n}size_{son_l})+size_{son_i}-j}\prod_{l=i+1}^{n}F(p_l,k))$$

同理，可以把括号内的化为 G 函数。

$$\sum_{i=1}^{k}\sum_{j=0}^{size_{son_i}-1}F(j,k)\cdot G((\sum_{l=i+1}^{n}size_{son_l})+size_{son_i}-j,k-i-1,k)$$

直接枚举计算是 $O(n^2)$。考虑 $k=2$ 时的优化方法，可以对 $size_{son_i}$ 与 $\frac{size_x}{2}$ 之间的关系分别讨论进行计算。
时间复杂度 $O(nk+n\log n)$，F 和 G 的推导需要用 Lagrange 反演。