计数专题

mazihang2022

2024.12.12

常用组合数恒等式

- $\sum_{i=k}^{n}\binom{i}{k}=\binom{n+1}{k+1}$。
- $\sum_{i=0}^{k}\binom{n}{i}\binom{m}{k-i}=\binom{n+m}{k}$。
- $\binom{n}{m}\binom{m}{k}=\binom{n}{k}\binom{n-k}{m-k}$，$m\binom{n}{m}=n\binom{n-1}{m-1}$。
- $\sum_{i=0}^{n}(-1)^i\binom{n}{i}=[n=0]$。
- $\sum_{i=0}^{n}\binom{n}{i}=2^n$，$\sum_{i=0}^{n}i\binom{n}{i}=n2^{n-1}$。
- 可以通过 $\binom{n}{m}=\binom{n-1}{m}+\binom{n-1}{m-1}$ 做整体递推。

组合意义

P8367 [LNOI2022] 盒

给定一个长为 n 的序列 a，可以花费 w_i 的代价将 a_i 加上 1、a_{i+1} 减去 1 或是将 a_i 减去 1、a_{i+1} 加上 1。需要保证操作之后 a 序列非负。对于所有满足 $\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} b_i$ 的非负序列 b，求出 a 变成 b 的最小代价之和。

$n \leq 5 \times 10^5$，$\sum_{i=1}^{n} a_i \leq 2 \times 10^6$。

组合意义

- 考虑 $(i,i+1)$ 操作的贡献，答案为：

$$\sum_{i=1}^{n-1} w_i \sum_{j=0}^{S} |s_i - j| f(i,j) f(n-i,S-j)$$

- 其中 $f(i,j)=\binom{i+j-1}{i-1}$ 表示 i 个不同盒子里放 j 个相同的球，盒子可以为空的方案数（插板法）。

组合意义

- 考虑 $(i,i+1)$ 操作的贡献，答案为：

$$\sum_{i=1}^{n-1} w_i \sum_{j=0}^{S} |s_i - j| f(i,j) f(n-i, S-j)$$

- 其中 $f(i,j) = \binom{i+j-1}{i-1}$ 表示 i 个不同盒子里放 j 个相同的球，盒子可以为空的方案数（插板法）。
- 核心在于对于所有 i 算出形如 $\sum_{j=0}^{k} f(i,j) f(n-i, S-j)$ 和 $\sum_{j=0}^{k} j f(i,j) f(n-i, S-j)$ 的式子。

组合意义

- 考虑 $(i,i+1)$ 操作的贡献，答案为：

$$\sum_{i=1}^{n-1} w_i \sum_{j=0}^{S} |s_i - j| f(i,j) f(n-i, S-j)$$

- 其中 $f(i,j) = \binom{i+j-1}{i-1}$ 表示 i 个不同盒子里放 j 个相同的球，盒子可以为空的方案数（插板法）。
- 核心在于对于所有 i 算出形如 $\sum_{j=0}^{k} f(i,j) f(n-i, S-j)$ 和 $\sum_{j=0}^{k} j f(i,j) f(n-i, S-j)$ 的式子。
- 其中可以通过分离变量得到 $jf(i,j) = j\binom{i+j-1}{i-1} = i\binom{i+j-1}{i}$。微调后变成前一种情况。

组合意义

- 对于 $\sum_{j=0}^{k} f(i,j)f(n-i,S-j)$ 考虑其组合意义：n 个不同的盒子放 m 个相同的球，前 i 个盒子放的球数不超过 k。转为枚举第 $k+1$ 个球的位置：$\sum_{j=0}^{k} f(i,j)f(n-i,S-j) = \sum_{j=i+1}^{n} f(j-1,k)f(n-j+1,S-k-1)$。注意到这个式子在 i 变化时能够快速维护。

斯特林数

- 第二类斯特林数：n 个不同的球放进 m 个相同的盒子的方案数。
- 递推：

$$\left\{ {n \atop m} \right\} = \left\{ {n-1 \atop m-1} \right\} + m \left\{ {n-1 \atop m} \right\}$$

- 方幂转下降幂：

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

组合意义

CF1097G Vladislav and a Great Legend

给定一棵树，记 $f(S)$ 表示点集 S 生成的虚树的边数。求出 $\sum_S f(S)^k$。
$n \leq 10^5$，$k \leq 200$。

组合意义

- 其实 k 次方可以直接通过二项式定理转移的！但是复杂度会带有 k^2。

组合意义

- 其实 k 次方可以直接通过二项式定理转移的！但是复杂度会带有 k^2。
- 转化方幂：

$$\sum_S f(S)^k = \sum_S \sum_{i=0}^{k} \left\{ {k \atop i} \right\} f(S)^{\underline{i}} = \sum_{i=0}^{k} i! \left\{ {k \atop i} \right\} \sum_S \binom{f(S)}{i}$$

- 考虑最后一项组合意义是在虚树中选出 i 条边，这样将该部分计入状态时，其状态数不超过子树大小，做树形背包复杂度就是 $O(nk)$ 的。

组合意义

- 其实 k 次方可以直接通过二项式定理转移的！但是复杂度会带有 k^2。
- 转化方幂：

$$\sum_S f(S)^k = \sum_S \sum_{i=0}^{k} \left\{ {k \atop i} \right\} f(S)^{\underline{i}} = \sum_{i=0}^{k} i! \left\{ {k \atop i} \right\} \sum_S \binom{f(S)}{i}$$

- 考虑最后一项组合意义是在虚树中选出 i 条边，这样将该部分计入状态时，其状态数不超过子树大小，做树形背包复杂度就是 $O(nk)$ 的。
- 状态设计：一种比较好的方式是 f_x 表示钦定虚树延伸到 x 的贡献。这样只需要在合并的时候计入答案就可以确保不重不漏。

无向连通图计数

- 求 n 个点的有标号无向图的数量。

无向连通图计数

- 求 n 个点的有标号无向图的数量。
- 枚举 1 号点所在连通块大小。

$$f_n = 2^{\binom{n}{2}} - \sum_{i=1}^{n-1} \binom{n-1}{i-1} f_i 2^{\binom{n-i}{2}}$$

欧拉图计数

- 求 n 个点的有标号且存在欧拉回路的图的数量。

欧拉图计数

- 求 n 个点的有标号且存在欧拉回路的图的数量。
- 条件：所有点度数为偶数且连通。

$$f_n = g_n - \sum_{i=1}^{n-1} \binom{n-1}{i-1} f_i g_{n-i}$$

- 其中 $g_n = 2^{\binom{n-1}{2}}$ 表示所有点度数为偶数的图的数量。

有向无环图计数

- 求 n 个点的有标号有向无环图的数量。

有向无环图计数

- 求 n 个点的有标号有向无环图的数量。
- 钦定入度为 0 的点的数量。

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

- 这里容斥的含义比起一般情况稍有不同，我们希望分配容斥系数使得 $\sum_{T \subseteq S} f(T) = 1$。

强连通图计数

- 求 n 个点的有标号强连通图的数量。

强连通图计数

- 求 n 个点的有标号强连通图的数量。
- 非强连通图缩点后是至少有两个点的有向无环图。

$$f_n = 2^{\binom{n}{2}} - \sum_{i=1}^{n-1} 2^{i(n-i)} 2^{\binom{n-i}{2}} \sum_{j=1}^{i} (-1)^{j+1} g_{i,j}$$

- 其中 $g_{i,j}$ 表示 i 个点组成了 j 个强连通图的方案数。

强连通图计数

- 求 n 个点的有标号强连通图的数量。
- 非强连通图缩点后是至少有两个点的有向无环图。

$$f_n = 2^{\binom{n}{2}} - \sum_{i=1}^{n-1} 2^{i(n-i)} 2^{\binom{n-i}{2}} \sum_{j=1}^{i} (-1)^{j+1} g_{i,j}$$

- 其中 $g_{i,j}$ 表示 i 个点组成了 j 个强连通图的方案数。
- 考虑整体转移 $h_n = \sum_{i=1}^{n} (-1)^{i+1} g_{n,i}$，换句话说带着容斥系数转移。

$$h_n = f_n - \sum_{i=1}^{n-1} \binom{n-1}{i-1} f_i h_{n-i}$$

强连通图计数

- 求 n 个点的有标号强连通图的数量。
- 非强连通图缩点后是至少有两个点的有向无环图。

$$f_n = 2^{\binom{n}{2}} - \sum_{i=1}^{n-1} 2^{i(n-i)} 2^{\binom{n-i}{2}} \sum_{j=1}^{i} (-1)^{j+1} g_{i,j}$$

- 其中 $g_{i,j}$ 表示 i 个点组成了 j 个强连通图的方案数。
- 考虑整体转移 $h_n = \sum_{i=1}^{n} (-1)^{i+1} g_{n,i}$，换句话说带着容斥系数转移。

$$h_n = f_n - \sum_{i=1}^{n-1} \binom{n-1}{i-1} f_i h_{n-i}$$

- 对应的题目是 UOJ37 【清华集训 2014】主旋律。

双射

- Prufer 序列和有标号无根树形成双射。
- 树对应到序列：取出编号最小的叶子结点删去，将其父亲加到序列末端。
- 序列对应到树：上述过程保证了点的度数等于其在序列中出现的次数加一。每次选择编号最小的当前状态下的叶子，其父亲唯一确定。

计数

- n 个点有标号无根树数量：n^{n-2}。
- n 个点且确定树上每个点度数后的有标号无根树数量：$\frac{(n-2)!}{\prod_{i=1}^{n}(d_i-1)!}$。
- n 个点形成 k 个连通块，添加 $k-1$ 条边使得原图连通的方案数：$n^{k-2}\prod_{i=1}^{k} s_i$。考虑一个点每在 Prufer 序列中出现一次就会有 s_i 种方案，答案实际上是 $(\sum_{i=1}^{k} s_i)^{k-2}\prod_{i=1}^{k} s_i$。

应用

CF917D Stranger Trees

给定一棵 n 个点的有标号树，对于 $1 \le i \le n$ 求出有多少棵 n 个点的树与给定的数恰有 i 条相同的边。
$n \le 100$。

应用

CF917D Stranger Trees

给定一棵 n 个点的有标号树，对于 $1 \le i \le n$ 求出有多少棵 n 个点的树与给定的数恰有 i 条相同的边。
$n \le 100$。

- $\prod s_i$ 拆成组合意义是经典的思路。

应用

P6596 How Many of Them

求有多少个 n 个点且割边不超过 m 条的无向连通图。

$n \leq 50$。

应用

P6596 How Many of Them

求有多少个 n 个点且割边不超过 m 条的无向连通图。
$n \leq 50$。

- 缩点之后所有割边构成一棵树，可以计数划分成若干个块的方案，然后用 Prufer 结论。

特殊容斥系数

[ABC236Ex] Distinct Multiples

给定 n, m 和长为 n 的序列 d。求有多少个长为 n 的序列 a 满足：
$1 \le a_i \le m$、$a_i \ne a_j$、$d_i \mid a_i$。
$n \le 16$，$m \le 10^{18}$。

特殊容斥系数

- 容易想到说考虑容斥，将原数组划分成若干组钦定组内元素相等再计数。问题在于这里的容斥系数还能是 $(-1)^k$ 吗？

特殊容斥系数

- 容易想到说考虑容斥，将原数组划分成若干组钦定组内元素相等再计数。问题在于这里的容斥系数还能是 $(-1)^k$ 吗？
- 实际上是不行的，因为本质上我们应该枚举 $\frac{n(n-1)}{2}$ 对不等关系中不被满足的，这时才是经典的容斥系数。

特殊容斥系数

- 容易想到说考虑容斥，将原数组划分成若干组钦定组内元素相等再计数。问题在于这里的容斥系数还能是 $(-1)^k$ 吗？
- 实际上是不行的，因为本质上我们应该枚举 $\frac{n(n-1)}{2}$ 对不等关系中不被满足的，这时才是经典的容斥系数。
- 考虑计算一种划分的容斥系数具体是多少，也就是所有恰形成当前划分的方案，其容斥系数之和。本质上只需要求出 $\sum_{T\subseteq E}(-1)^{|T|}[T \text{ is connected}]$。

特殊容斥系数

- 容易想到说考虑容斥，将原数组划分成若干组钦定组内元素相等再计数。问题在于这里的容斥系数还能是 $(-1)^k$ 吗？
- 实际上是不行的，因为本质上我们应该枚举 $\frac{n(n-1)}{2}$ 对不等关系中不被满足的，这时才是经典的容斥系数。
- 考虑计算一种划分的容斥系数具体是多少，也就是所有恰形成当前划分的方案，其容斥系数之和。本质上只需要求出 $\sum_{T\subseteq E}(-1)^{|T|}[T \text{ is connected}]$。
- 用图计数相关技巧，考虑减去不连通图的贡献：

$$
\begin{aligned}
f_n &= g_n - \sum_{i=1}^{n}\binom{n-1}{i-1}f_i g_{n-i} \\
&= [n=1] - \sum_{i=1}^{n-1}\binom{n-1}{i-1}f_i[n-i=1] \\
&= [n=1] - (i-1)f_{i-1}
\end{aligned}
$$

点边容斥

XYD341 树的重心

给定一棵 n 个点的树，可以断掉其中若干条边，此时的权值定义为所有连通块重心的编号和（若有两个重心则都计入权值）。求所有不同的断边方案的权值和。

$n \leq 3 \times 10^3$。

点边容斥

- 考虑重心的判定方式。我们选择：重心是最深的满足 $sz_x \geq \lceil \frac{n}{2} \rceil$ 的点。

点边容斥

- 考虑重心的判定方式。我们选择：重心是最深的满足 $sz_x \geq \lceil \frac{n}{2} \rceil$ 的点。
- 我们注意到满足 $sz_x \geq \lceil \frac{n}{2} \rceil$ 的点形成一条祖先后代链，因此可以考虑类似点边容斥的做法：将点 x 的权值设为 $x - fa$，条件改为 $sz_x \geq \lceil \frac{n}{2} \rceil$ 就将其权值计入。

点边容斥

- 考虑重心的判定方式。我们选择：重心是最深的满足 $sz_x \geq \lceil \frac{n}{2} \rceil$ 的点。
- 我们注意到满足 $sz_x \geq \lceil \frac{n}{2} \rceil$ 的点形成一条祖先后代链，因此可以考虑类似点边容斥的做法：将点 x 的权值设为 $x - fa$，条件改为 $sz_x \geq \lceil \frac{n}{2} \rceil$ 就将其权值计入。
- 状态设计：关键问题在于断边操作导致 n 不是固定的，如何处理上面的条件？

点边容斥

- 考虑重心的判定方式。我们选择：重心是最深的满足 $sz_x \geq \lceil \frac{n}{2} \rceil$ 的点。
- 我们注意到满足 $sz_x \geq \lceil \frac{n}{2} \rceil$ 的点形成一条祖先后代链，因此可以考虑类似点边容斥的做法：将点 x 的权值设为 $x - fa$，条件改为 $sz_x \geq \lceil \frac{n}{2} \rceil$ 就将其权值计入。
- 状态设计：关键问题在于断边操作导致 n 不是固定的，如何处理上面的条件？
- 条件可以改写为父亲方向的点数不超过当前点子树大小，可以分两阶段 DP。

树的拓扑序计数

- 树的拓扑序数量：$\frac{n!}{\prod_{i=1}^{n} sz_i}$。

树的拓扑序计数

- 树的拓扑序数量：$\frac{n!}{\prod_{i=1}^{n} sz_i}$。

树上排列计数

给定一棵 n 个点的树，对于一个排列 a，记 k 表示有多少个点 x 满足它的所有祖先 y 都有 $a_x \le a_y$，定义这个排列的权值为 m^k，m 是给定的常数。求所有排列的权值和。

$n \le 5 \times 10^3$。

树的拓扑序计数

- 钦定若干个点满足条件，这些点的大小关系限制构成一棵树，而对于剩下不满足条件的点 x，你向上找到第一个满足条件的点 y，要求 $v_x \geq v_y$。

树的拓扑序计数

- 钦定若干个点满足条件，这些点的大小关系限制构成一棵树，而对于剩下不满足条件的点 x，你向上找到第一个满足条件的点 y，要求 $v_x \geq v_y$。
- 这并不等价于树的拓扑序计数，因为边的方向并不统一，不过容斥一下就好了。

树的拓扑序计数

- 钦定若干个点满足条件，这些点的大小关系限制构成一棵树，而对于剩下不满足条件的点 x，你向上找到第一个满足条件的点 y，要求 $v_x \geq v_y$。
- 这并不等价于树的拓扑序计数，因为边的方向并不统一，不过容斥一下就好了。
- 状态设计：只需要记录当前容斥后限制所构成树的大小，我们在每次钦定一个点不满足条件时就做一步容斥，而不是等碰到满足条件点再容斥。

延后钦定

MEX counting

给定 n, k 和长为 n 的序列 b，求有多少个序列 a 满足 $0 \le a_i \le n$ 且 $|\operatorname{mex}(\{a_1, a_2, \ldots, a_i\}) - b_i| \le k$。
$n \le 2 \times 10^3$，$k \le 50$。

延后钦定

- 记录当前 mex，再记录在这之前有多少种 $>$ mex 的数。只在当前 mex 变化的时候确定之前的数具体是什么。

延后钦定

- 记录当前 mex，再记录在这之前有多少种 $>$ mex 的数。只在当前 mex 变化的时候确定之前的数具体是什么。
- 相比记录之前有多少个 $>$ mex 的数的好处：这样记录需要在转移时去做钦定，枚举的东西多了一维，而变加入边钦定就不用了。

连续段 DP

- 其实是在做“插入”形式的 DP。核心是状态设计：形成了 i 个连续段意味着，段内不会再插入数，段间还需要插入数。

连续段 DP

- 其实是在做“插入”形式的 DP。核心是状态设计：形成了 i 个连续段意味着，段内不会再插入数，段间还需要插入数。

[TopCoder11213] Apple Trees

考虑数轴上坐标在 $1\sim d$ 内的点，有 n 棵树要种在这些点上，要求第 i 棵树与两旁相邻的树的距离至少为 r_i，问种树的方案数。$d\le 10^5$，$n,r_i\le 40$。

连续段 DP

- 坐标范围较大，考虑从最紧的状态用插板法得到最终方案数。

连续段 DP

- 坐标范围较大，考虑从最紧的状态用插板法得到最终方案数。
- 按 r_i 从小到大考虑，使用连续段 DP，$f_{i,j,k}$ 表示考虑前 i 棵树形成 j 个连续段总长度为 k 的方案数，一共三种转移。

连续段 DP

CF1515E Phoenix and Computers

有 n 台电脑排成一排，最初所有电脑都是关着的。你可以手动开启一台电脑，在任意时刻，若电脑 $i-1$ 和电脑 $i+1$ 都已经开启，电脑 i 会被自动开启。问有多少种开启所有电脑的方式，方式不同当且仅当你的操作不同。

$n \leq 400$。

连续段 DP

CF1515E Phoenix and Computers

有 n 台电脑排成一排，最初所有电脑都是关着的。你可以手动开启一台电脑，在任意时刻，若电脑 $i-1$ 和电脑 $i+1$ 都已经开启，电脑 i 会被自动开启。问有多少种开启所有电脑的方式，方式不同当且仅当你的操作不同。
$n \leq 400$。

- 依然考虑连续段 DP。$f_{i,j}$ 表示已开启 i 台电脑构成了 j 个连续段，每段间距离大于 1，此时对应的方案数。一共五种转移。

状态设计

CF1781F Bracket Insertion

定义在括号串 s 上的一次操作：选择一个空位，以 p 的概率插入 ()，以 $1-p$ 的概率插入)(。最初 s 为空，求出 n 次操作后得到合法括号序列的概率。

$n \leq 500$。

状态设计

- 考虑第一次操作，之后变为子问题。

状态设计

- 考虑第一次操作，之后变为子问题。
- 假设插入了 ()，这样会把原序列分为三个区间，要求左右区间本身合法，中间区间合法条件稍宽松一点：前缀和可以到 -1，这样的话会是子问题：$f(l,r,x)$ 表示区间 l,r，前缀和最小可以到 $-x$ 的方案数。答案只和区间长度有关于是不用同时记 l,r。直接枚举转移是 $O(n^4)$ 的，但这种三段式的转移可以考虑设出辅助数组分步转移。

贡献

CF1842G Tenzing and Random Operations

给定长为 n 的序列 a 和一个整数 v。对该序列进行 m 次操作。每次操作中，等概率随机选择整数 $i \in [1, n]$，然后对于所有 $j \in [i, n]$ 将 a_j 加上 v。求出操作完后 $\prod_{i=1}^{n} a_i$ 的期望值。
$n \le 5 \times 10^3$，$m, v, a_i \le 10^9$。

贡献

- 考虑用分配律拆开连乘。

贡献

- 考虑用分配律拆开连乘。
- 这样的话我们只需要对于每个 i 选出 a_i 或 v 其中之一计算贡献即可，然后选 v 的时候我们钦定一下这是哪一次操作带来的 v。设 $f_{i,j}$ 表示前 i 个数钦定了 j 次操作的贡献，这里 $j \leq n$ 因为如果没有选择一个 v 做出贡献我们就不用考虑对应具体是怎么操作的。

对应、双射

CF1264D2 Beautiful Bracket Sequence (hard version)

对于一个括号序列，若其由 x 个左括号和 x 个右括号按顺序拼接形成，则定义其深度为 x，否则深度被定义为 0。
对于一个括号序列，定义其权值为删除任意多字符后得到的深度最大的括号序列的深度。
给定一个包含左右括号和问号的序列，问号可以任意替换为左括号或右括号，问替换后所有可能的括号序列的权值和。
$n \leq 10^6$。

对应、双射

- 一个括号串的权值为 $\max p_i + s_{i+1}$。
- 注意到 i 变大 1 的时候，p 变大 1、s 减小 1 恰有一者成立，那么一定存在唯一的一个位置 t 使得 $\max(p_t, s_{t+1})$ 最大且 $p_t = s_{t+1}$，在这里统计就好。

状态优化

AGC017F Zigzag

给定一个 n 层的三角形图，第 i 层有 i 个点，标号为 $(i,1),(i,2),\ldots,(i,i)$。你需要从 $(1,1)$ 向下画 m 条折线，对于每一个小段，你可以从 (i,j) 画到 $(i+1,j)$ 或者 $(i+1,j+1)$。你需要保证第 i 条折线整体在第 $i-1$ 条折线的右侧。
此外，有 k 条额外的限制，每条限制形如 (a_i,b_i,c_i)，表示第 a_i 条折线处于位置 (b_i,j) 时，下一小段必须画向 $(b_i+1,j+c_i)$。问有多少种画折线的方案数，对 10^9+7 取模。
$n,m\leq 20$。

状态优化

- 从左到右转移折线，逐格转移。这样我们需要记录上一条折线的后缀和当前的折线，还需要记录上一条折线在当前行的位置。考虑怎么优化状态。

状态优化

- 从左到右转移折线，逐格转移。这样我们需要记录上一条折线的后缀和当前的折线，还需要记录上一条折线在当前行的位置。考虑怎么优化状态。
- 假设前一条线二进制表示是 A 当前线是 B，发现限制可以描述为 B 的前缀 popcount 大于 A 的前缀 popcount，那么你当前在填 B 的过程中，填一个 1 就消掉 A 中的一个 1，这样去表示限制，就不用多记状态了。这个在图上的意义是转移一步时将当前折线还能到的地方和上一条折线的限制取交集，该交集最上方的点一定就在当前折线所处的位置，以此来描述新的限制。

格路计数

- 单纯计数时常考虑沿直线翻折。
- 可以把一些问题转化为格路计数，常见的转化是括号串。

格路计数

- 单纯计数时常考虑沿直线翻折。
- 可以把一些问题转化为格路计数，常见的转化是括号串。

CF1924D Balanced Subsequences

求有多少个由 n 个左括号 m 个右括号组成的，最长合法括号子序列长度恰为 $2k$ 的括号串。
$n, m, k \le 2 \times 10^3$。

格路计数

- 单纯计数时常考虑沿直线翻折。
- 可以把一些问题转化为格路计数，常见的转化是括号串。

CF1924D Balanced Subsequences

求有多少个由 n 个左括号 m 个右括号组成的，最长合法括号子序列长度恰为 $2k$ 的括号串。
$n, m, k \leq 2 \times 10^3$。

- 等价于最低点恰等于左括号数量。

更多技巧

- min-max 容斥。
- 期望的线性性。
- 阶梯化贡献：$x=\sum_{i=0}^{n}[x>i]$。
- 矩阵树定理：度数矩阵减邻接矩阵。
- BEST 定理：$cnt\prod_{i=1}^{n}(d_i-1)!$，$cnt$ 表示内向树数量。
- Polya：$\frac{1}{|G|}\sum_{p\in G}X^g$。
- DP 套 DP。
- ……
- 随机丢一点题，可以看看对应到了哪些技巧：P7213、CF1909F2、ARC163D、AGC024E。