DP 计数

范斯喆

浙江省诸暨市海亮高级中学

2025 年 4 月 22 日

前言

本次讲课的主题是 DP 计数。
众所周知，在思考计数问题时，可能会遇到两种问题：数重或者数漏。这些问题的本质原因，是对计数对象没有较好的刻画。
我将在这里分享常见的刻画计数对象的几种手法。各种手法之间的界限可能并不是很清晰，大家以理解为主，勿生搬硬套。
本人做题不多，因此会用一些很经典的题进行举例。如果您做过忽略即可。

推导合法条件

第一种方法是对合法条件进行讨论和变形，转化为对更简单的限制，也就是所谓的“推性质”。可能的思考方向有：

- 从合法条件出发进行推导，得到与其等价的条件；
- 模拟小样例或者有特殊性质的样例，尝试找到刻画方式；
- 想象一个合法的对象，枚举它有哪些可能性；
- 联想一些与题目条件相似的模型；
- ……

得到更方便描述和计算的条件后，就可以开始尝试 DP。当然，也有很多题目到这一步后就可以直接用式子描述了，不需要 DP。

[省选联考 2025] 封印

给定一个序列 $a_1, a_2, \cdots, a_n$，每个元素都是 1 到 m 之间的正整数。
你可以对序列进行任意次修改（可以 0 次）。每次选择序列的某个**严格前缀最大值** a_x。若 $a_x > 1$，则将 $a_x - 1$ 插入序列 a 的最后。然后把 a_1 到 a_x 全部删除。
求你可能得到多少种本质不同的非空序列。
$1 \leq n, m \leq 2500$。

题解

设序列的最大值为 mx。在操作到序列的第一个最大值之前，“严格前缀最大值”个数一定只会减少而不会增加。那么不妨把这个位置视为一个分割点。

操作完这个位置后，下一个 mx 出现的位置又会成为分割点。把 mx 删完后，后面所有 $mx-1$ 出现的位置就是分割点。

现在序列被分割点分为若干部分，且所有分割点除非删到 0 否则是不可能删掉的。

那么考虑两个分割点之间的段可能被删成什么形态。通过手玩发现，如果把每个数向它前面最近的不小于它的数连边（如果没有，则向虚拟根连边），则一定是删掉这棵树的若干个子树，保留一个包含根的连通块。这是十分容易计数的。

题解

于是，当第一个分割点被操作两次之后，每两个分割点之间的序列就互相独立了。其中有一个会跨越序列首尾，也是容易处理的。
现在我们会了操作充分多次的情况，如果操作的太少，以至于原序列的 n 个元素还没有删完呢？这也可以用简单的 DP 求出。
然后就只剩下从“原序列被删完”到“第一个分割点被操作第二次”之间的情况。这和之前说的情况类似。
复杂度显然是 $O(n^2)$。细节比较多。

[省选联考 2023] 染色数组

考虑一个正整数序列 $a_1, a_2, \cdots, a_n$，值域为 $[1, m]$。
你要将每个数染成红色或绿色，满足红色的数递增，绿色的数递减。如果一个序列存在**至少两种**染色方案，则称它是**完美的**。
定义一种给染色方案打分的方式。对于每个 $i < j$：

- 如果 a_j 被染成红色，且 $a_j < a_i$，则得 $m - a_j + 1$ 分；
- 如果 a_j 被染成绿色，且 $a_j > a_i$，则得 a_j 分；
- 否则不得分。

定义一个完美序列的得分为它的染色方案的得分最大值。
给定序列的前 t 个数，你需要回答：

1. 有多少种确定后 $n - t$ 个数的方案，使得形成一个完美数组？
2. 所有可能的完美数组的得分之和是多少？

$1 \le n \le 50$，$1 \le m \le 200$。

题解

首先考虑怎样的序列是完美的。

一个序列两种不同的染色方案，则必然存在某个数可以染红色也可以染绿色。稍加推导得知，这样的数 a_i 需要满足：

- $[1, i)$ 中 $\le a_i$ 的数和 $(i, n]$ 中 $\ge a_i$ 的数递增；
- $[1, i)$ 中 $\ge a_i$ 的数和 $(i, n]$ 中 $\le a_i$ 的数递减；

不妨称这样的 a_i 为**关键位置**。

下面考察关键位置的性质。这种若干个关键点/关键位置可能形成一个子序列，而子串肯定是比子序列更好处理的，所以这种问题中我们肯定有一个想法就是确定关键点连续段的性质。

事实上，如果存在某两个关键位置 l, r，则：

- 若 $a_l = a_r$，则必须 $r - l = 1$，否则中间的数肯定不合法；
- 若 $a_l < a_r$，则中间的数必须也在 (a_l, a_r) 之间，而且它们也会是关键位置；
- $a_l > a_r$ 的情况类似。

题解

现在我们知道关键位置一定是一个区间。而且，要么区间内的数有严格单调性，要么区间长度等于 2。
现在考察非关键位置的形态。关键位置区间为 $[l, r]$，$L = \min(a_l, a_r)$，$R = \max(a_l, a_r)$。则有：

- $[1, l)$ 中 $> R$ 的递减，$< L$ 的递增，不存在 $[L, R]$ 中的元素；
- $(l, n]$ 中 $> R$ 的递增，$< L$ 的递减，不存在 $[L, R]$ 中的元素。

现在可以解决第一问。枚举 r 和 a_r。

- 如果是 $a_{r-1} = a_r$，可以分别求出 $[1, r-1)$ 和 $(r, n]$ 的方案数。
- 否则枚举 a_{r+1}。显然 $a_r \neq a_{r+1}$，不妨设 $a_r < a_{r+1}$。后缀直接算，前缀用容斥做一下即可。

现在问题变为，给定 r, x, y，求 $[1, r)$ 中 $> y$ 的递减、$< x$ 的递增的方案数。这是可以 DP 求出的。

题解

接下来是第二问，显然我们想找到最优的染色方案。
还是设关键位置为 $[l, r]$。不在这个区间内的元素的染色都可以唯一确定，考虑区间内的元素是如何染色的。

- 如果 $r - l = 1$，l 和 r 必须一红一绿，方案得分相等。
- 如果 $a_l < a_r$，则这里只有至多一个元素是绿色，其他都是红色。因为绿色元素越大越好，所以一定是 a_r 绿色，其余红色；
- 如果 $a_l > a_r$，同理。

后面的计数太复杂了，而且全是无脑分讨，所以我们不讲了。

[IOI 2020] 装饼干

有 k 种物品，编号为 $0, 1, \cdots, k-1$。第 i 种物品有 a_i 块，价值为 2^i。
你需要将饼干装袋。给定 x，求有多少个 y，使得可以凑出至少 x 袋价值总和为 y 的饼干。
$1 \leq k \leq 60$。多组数据，数据组数不超过 1000。

题解

从低位到高位考虑，得到合法的充要条件为

$$\forall 0 \le i < 60, \sum_{j \le i} 2^j y_j \le \frac{\sum_{j \le i} 2^j a_j}{x}$$

然而，如果从低位往高位 DP，就要记录还剩多少和，很难优化。改为从高位往低位考虑。这样也能给出一个充分性的构造，也就是贪心。

枚举 y 的最高位 i，从它开始往下贪心拼 x 个 2^i。设最后用到了下标为 j 的，剩余 a'_j。

题解

按照 a'_j 分类讨论：

- 如果 $a'_j > x$，这就意味着原来的 $a_j \geq x+2$，那可以一开始就合并两个 2^j。所以不用考虑这种情况；
- 如果 $a'_j = x$，那么 y 中的次高 1 位至多是 $j+1$。并且如果次高 1 位 $\leq j$，那个 i 实际上根本没影响；
- 如果 $a'_j < x$，那么 y 中的次高 1 位至多是 j。并且如果次高 1 位 $< j$，那个 i 也根本没影响。

所以，如果设 f_i 表示只考虑低 i 位且 $y_i = 1$ 的答案，则 f_i 的转移形如 $\sum_{k<j} f_j$ 加上一个东西，这个东西可以递归计算。
用前缀和算那个 σ，复杂度 $O(k^2)$。

CF1603E A Perfect Problem

称一个序列为**好的**当且仅当它的 $\max \times \min \geq \text{sum}$。
给定 n，求长度为 n、每个数在 $[1, n+1]$ 范围内，且每个**非空子序列**都是好序列的序列个数。
$1 \leq n \leq 200$。

题解

分析一个序列合法的条件。先排序。
不难看出，每个子序列是好的，等价于每个子串是好的（因为 $\min$ 和 $\max$ 不变，sum 增大），又等价于每个前缀是好的（因为 $\max$ 不变，$\min$ 减少，sum 增大）。
显然有 $a_i \geq i$。且如果 $a_i = i$，必须有 $\forall 1 \leq j \leq i, a_j = i$。否则 $[1, i]$ 就不合法。
观察到值域是 $[1, n+1]$，不难想到分 $a_n = n$ 和 $a_n = n+1$ 讨论。
前者很简单，后者需要继续讨论。

题解

如果 $a_n = n+1$，考虑一个 $a_i \geq i+1$ 的位置。可知，如果 $a_{1\sim n}$ 合法，则 $a_{1\sim i}$ 合法。这是因为

$$(n+1)a_1 \geq \sum_{j=1}^{n} a_j \Rightarrow a_i a_1 \geq (i+1)a_1 \geq \sum_{j=1}^{i} a_j$$

枚举 a_1，a 合法的充要条件为：

- $\forall 1 \leq i \leq a_1, a_1 \leq a_i \leq n+1$；
- $\forall a_1 < i \leq n, i+1 \leq a_i \leq n+1$；
- $(n+1)a_1 \geq \sum_{i=1}^{n} a_i$。

然后就能 $O(n^3 \log n)$ DP 了。

又因为 $a_1 < n - O(\sqrt{n})$ 时不可能满足最后一个条件，所以复杂度为 $O(n^3\sqrt{n}\log n)$。

对判定过程 DP

这种方法其实是第一种方法的子类。
有时候，当我们推导合法条件时，可能并不能直接得到合法需要满足的性质，而是得到判定合法的“过程”。
这种“过程”有时候是贪心，有时候是 DP。如果是 DP，就是我们常说的“DP 套 DP”。
这个“过程”也可能可以写成式子的形式，比如前面“装饼干”那题的式子就是通过从小到大填数的过程列出的。

[PA 2021] Od deski do deski

称一个序列为**好的**当且仅当通过“删除一个长度至少为 2 且两端相等的区间”可以在若干次操作后被删空。
给定 n, m，求长度为 n，每个元素在 $[1, m]$ 之间的好序列个数。
$1 \leq n \leq 3000$，$1 \leq m \leq 10^9$。

题解

考虑序列确定的情况，尝试判断能不能删空。显然可以 DP，设 f_i 表示 $[1,i]$ 能不能删空，则

$$f_i = \bigvee_{j=1}^{i-1} [a_i = a_j] f_{j-1}$$

这样判定是 $O(n^2)$ 的，但其实我们只要记录 $S = \{a_j \mid f_{j-1} = 1\}$ 即可。
接下来思考如何计数，我们注意到状态里只要记录 $|S|$ 就行了。
状态 $O(n^2)$，转移 $O(1)$。

[AGC022E] Median Replace

有一个长度为 n 的 01 串 s，其中有若干个位置还未确定。
一次操作可以将 3 个连续的字符变为它们的中位数。
求有多少确定整个串的方案，使得进行 $\frac{n-1}{2}$ 次操作后可以剩下 1。
$1 \leq n \leq 3 \times 10^5$。保证 n 是奇数。

题解

考虑判定。显然可以贪心。用一个栈维护当前状态，栈顶是一段 0，栈底是一段 1。
加入 0 时，如果当前栈里有两个 0 就可以消掉，否则先往栈里放。
加入 1 时，如果栈顶为 0，则这个 1 可以和 0 抵消。否则如果栈里有两个 1，就忽略。如果为空或者只有一个 1，放进去。
最后只需要栈里 1 的个数不小于 0 的个数即可。
不难发现栈里的 $0/1$ 个数均 ≤ 2。那么做一个 9 个状态的自动机上 DP 即可。
复杂度 $O(n)$。

Median Operations

给定一个长度为 n 的排列。
一次操作可以将 3 个连续的数变为它们的中位数。
求对于每个 $k = 1, 2, \cdots, n$，是否可以通过 $\frac{n-1}{2}$ 次操作，让序列剩下的元素变为 k。
$1 \leq n \leq 2 \times 10^5$。保证 n 是奇数。

题解

枚举 k，合法的充要条件是可以剩下 $\leq k$ 的数也可以剩下 $\geq k$ 的数。分别赋值 0 或 1 并判定即可。
k 增大的时候是单点修改，用线段树维护转移。
复杂度 $O(n \log n)$。

[NOI2023] 桂花树

给定一棵 n 个节点的树 T，保证 T 的非根节点的父亲编号小于自己。
给定整数 k，称一棵 $n+m$ 个节点的树 T' 是**好的**，当且仅当它满足以下所有条件：

1. 对于所有满足 $1 \le i, j \le n$ 的 (i, j)，在树 T 和 T' 上，i 和 j 的最近公共祖先相同。
2. 对于所有满足 $1 \le i, j \le n+m$ 的 (i, j)，在树 T' 上，i 和 j 的最近公共祖先的编号不超过 $\max(i, j) + k$。

求有多少棵 T' 是好的。
$1 \le n \le 3 \times 10^4$，$1 \le m \le 3 \times 10^3$，$0 \le k \le 10$。

题解

考虑如何判定一个树是否是合法的。看到 $\max(i,j)$ 我们不难想到从小往大插入所有点，并维护当前所有点的虚树。
那么限制就是，加入 $1\sim i$ 的点后，虚树上的点的编号均 $\leq i+k$。
加入一个点时，如果加在某条边中间分出去的子树中，会在这条边中间创建一个新点。状压一下哪些点创建了新点即可。
复杂度 $O(mk2^k)$。

[AGC064D] Red and Blue Chips

有 n 个字符串，初始每个字符串都只有一个字符 R 或 B。保证第 n 个是 B。
对于每个 $i = 1, 2, \cdots, n-1$，你需要选择一个整数 j 满足 $i < j \leq n$，且第 j 个字符串的最后一个字符是 B，然后把第 i 个字符串拼在第 j 个的前面。
问最后能得到多少种本质不同的字符串。
$2 \leq n \leq 300$。

考虑判定一个结果串能否被得到。
从后往前考虑。最后一个 B 之前的连续 R 段必须在序列的最前面一段。考虑下一个 B 之前的连续 R 段，它们可以选择放在这个 B 的前面或者后面。之后的 R 同理。
那么现在就有一个判定方案：

- 设一个计数器 x，初始为结果串最左边一段 R 的长度。
- 从右到左扫描原序列，每当扫描到一段极长的 R 时，我们先从 x 中减去这段 R 的长度，再从结果串中选择未被选择的最长的 R 段，将 x 加上那一段的长度。

先枚举结果串第一段的长度，然后进行 DP，从大到小加入那些 R 段即可。
因为调和级数的存在，复杂度为 $O(n^3 \log n)$。

[AGC069D] Tree and Intervals

给定 n。
对于所有 n 个节点的树，定义一个长度为 $n-1$ 的序列 x。初始时 x 全为 0。对于树的每条边 (u,v)，将所有满足 $\min(u,v) \le i < \max(u,v)$ 的 x_i 加上一。
求本质不同的 x 序列的数量。
$2 \le n \le 500$。

题解

已知一棵树，如何计算 x 序列？
显然是按编号从小到大染黑所有点，x_i 等于加入了 $1 \sim i$ 之后的同色连通块数。
考虑在维护同色连通块数的同时，维护黑色连通块数。那么：

- 同色连通块数增加，则黑色连通块数要么 $+1$，要么不变；
- 同色连通块数不变，黑色连通块数也不变；
- 同色连通块数减少，假设同色连通块数减少了 cnt，则黑色连通块数要么 $-cnt$，要么 $-(cnt-1)$。要注意不能让同色连通块数和黑色连通块数相同，除非是最后一次操作。

显然我们会让黑色连通块数尽可能大，这样不容易减到 ≤ 0。
设 $dp_{i,j,k}$ 表示填了 $1 \sim i$，黑色连通块数为 j，总连通块数为 k 的方案数。转移可以用前缀和优化，做到 $O(n^3)$。

CF924F Minimal Subset Difference

定义 $f(n)$ 表示将 n 的所有数位之间插入加号或减号，最终得到的值的绝对值最小值。
T 组询问，每次给定 l, r, k，求 $l \le n \le r$ 且 $f(n) \le k$ 的 n 的个数。
$1 \le T \le 5 \times 10^4$，$1 \le l \le r \le 10^{18}$，$0 \le k \le 9$。

题解

如果给定 n，那这就是一个背包问题。
背包上界只需开到 $\frac{18\times 9+9}{2}$ 即可。
DP 套 DP。先搜出状态，再跑个 DFA 最小化随便过。

[USACO22FEB] Phone Numbers P

有一个九键手机。你想要打出一个目标序列，为了节省时间，你可以用以下三种手法：

1. 按下某个键；
2. 同时按下相邻的两个键；
3. 同时按下组成一个正方形的四个键。

当你同时按下若干个键时，这些键的输入顺序是随机的。当然，你的输入方式必须可能打出目标序列。
给定你实际输入的序列 S，求有多少种可能的目标序列。
$1 \leq |S| \leq 10^5$。

题解

判定是好做的，设当前判定到 i，只需记录 $i-3$，$i-2$，$i-1$，i 这四个前缀能否被打出就行了。
如果要计数，需要额外记录 $i-2$，$i-1$，i 这三位的数字。
还是跑 DFA 最小化。

???

有一个九键手机。你想要打出一个目标序列，为了节省时间，你可以用以下三种手法：

1. 按下某个键；
2. 同时按下相邻的两个键；
3. 同时按下组成一个正方形的四个键。

当你同时按下若干个键时，这些键的输入顺序是随机的。当然，你的输入方式必须可能打出目标序列。

给定两个包含 $0 \sim 9$ 和 ? 的序列 S_1, S_2。求有多少种将把 ? 替换为数字的方式，使得有可能想输入 S_1，打出了 S_2。

$1 \leq |S| \leq 10^5$。

题解

这和上一题真的一模一样……

固定单射

本来叫做钦定单射。
这种方法主要用来解决“本质不同”的问题。思路是在每个“本质相同”的类中选择一个作为代表元素，也就是固定一个从类到其中元素的单射。
选择方法有很多种。
这种方法和容斥有点像，都是对每个计数对象固定一个系数。本质上它们是“带权计数”的两种特殊情况：固定单射是有一些系数为 1，另一些系数为 0，容斥是系数为 1 或 -1。
当然，存在更复杂的带权计数。不过通常可以用其他方式来理解。

CF1679F Formalism for Formalism

给定正整数 n，以及 m 组无序数对 (u_i, v_i)。
若一个数字的相邻两位数 (x, y) 在这些无序数对中出现过，则可以交换 x 和 y。若一个数 A 可以通过若干次交换操作得到 B，则称 A 和 B 是等价的。
求 $[0, 10^n - 1]$ 中等价类的数量。如果数字不足 n 位，则补前导零。
$1 \le n \le 5 \times 10^4$，$0 \le m \le 45$，$0 \le u_i, v_i \le 9$。

题解

把每个等价类映射到其中最小的数，转化为数“在它的等价类里最小”的数的个数。
显然，“在它的等价类里最小”等价于：对每个位置 x，x 往前能移到的一段区间中不存在 $> x$ 的数字。
由于数字只有 10 个，所以用状压记录每个数字能不能填即可。
复杂度 $O(n2^{10}10)$。

CF1740F Conditional Mix

给定一个长度为 n 的序列 $a_1, a_2, \cdots, a_n$。
有 n 个集合，一开始每个集合只有一个元素 a_i。可以操作任意次，每次可以合并两个交集为空的集合。
设合并完之后每个集合的元素个数构成可重集 S，求 S 的种类数。
$1 \leq n \leq 2 \times 10^3$，$1 \leq a_i \leq n$。

题解

先求出 i 在 a 中的次数 cnt_i。
把 S 从大到小排序，补 0 补到长度为 n。设这个序列为 s_i。
考虑怎么判定 S 是否合法。从前往后枚举 i，我们要凑出一个大小为 s_i 的集合。那我们肯定优先用 cnt 大的数。这样就把 S 对应到了唯一的一个合并方案。
分析一下，其实条件就是

$$\sum_{j=1}^{i} s_j \leq \sum_{j=1}^{n} \min(i, cnt_j)$$

那么从前往后用背包做即可。时间复杂度 $O(n^2 \log n)$。

[AGC008F] Black Radius

有一棵 n 个节点的树，一开始每个节点均为白色。其中有些点是好的，另一些点是不好的。
你要选择一个好的点 x，然后选择一个自然数 d，将所有与 x 距离不超过 d 的点都染成黑色。
问染色一次后，有多少种可能的状态。
$2 \leq n \leq 2 \times 10^5$。

题解

考虑将一种状态唯一地对应一对 (x, d)。不妨令其为 d 最小的。但是 d 最小的方案可能有多种。一种情况是整棵树都被覆盖；另一种情况是 d 最小对应的 x 不好，这种情况下我们就视为可以选 x。

那么枚举 x。先分析 x 是好的的情况。对于每个与 x 相邻的点 y，要么以 y 为重心不能覆盖黑色区域，要么 d 比以 x 为重心的时候大。

可以选的 d 只受到两个限制：

1. 不能覆盖整棵树；
2. 对于每个 y，以 y 为中心，$d-1$ 为半径的区域不能覆盖 x 的其它子树。

这用换根 DP 就能算。

题解

对于 x 不好的情况，我们对 d 有一个下界限制。
具体地，必须存在至少一个与 x 相邻的点 y，使得 y 子树内有好点，并且这个方案覆盖了 y 的整棵子树。充要性很容易证明。
所以只要换根 DP 就能 $O(n)$ 求解。

[集训队互测 2022] Range Minimum Element

有一个长度为 n，值域为 $[1, c]$ 的正整数序列 a。
给定 m 个区间 $[l_i, r_i]$，定义长度为 m 的序列 b 满足 $b_i = \min_{j=l_i}^{r_i} a_j$。
求 a 任意的情况下，有多少种可能的 b。
$1 \le n \le 100$，$1 \le m \le \frac{n(n+1)}{2}$，$1 \le c \le 998244353$。

题解

考虑将一组 b 对应到一个 a，令 a_i 为包含它的区间的 b 的最大值即可。那么我们对这样的 a 进行计数就行了。

这似乎不是很好计数。从大到小枚举 x，观察所有 $\geq x$ 的 a_i。它们之间如果有空隙，就一定还没被区间覆盖过。因此这些已经确定的连续段之间互相独立。

那么设 $dp_{x,l,r}$ 表示填了 $[x,c]$，恰好填满区间 $[l,r]$ 的方案数。每枚举一个 x，先枚举 l，然后从前往后枚举 r 并转移即可。

要枚举 c 层，每层做一个 $O(n^3)$ 的 DP，复杂度为 $O(n^3c)$。显然答案是关于 c 的多项式，因此可以拉插，优化到 $O(n^4)$。

[AGC056B] Range Argmax

有一个长度为 n 的排列 a。
给定 m 个区间 $[l_i, r_i]$，定义长度为 m 的序列 b，满足 b_i 是 $a_{l_i}, a_{l_i+1}, \cdots, a_{r_i}$ 中最大值对应的下标。
求 a 任意的情况下，有多少种可能的 b。
$1 \leq n \leq 300$，$1 \leq m \leq \frac{n(n+1)}{2}$。

题解

同样地，考虑将一组 b 对应到一个 a。从大到小枚举 i，把 i 放到**最左侧的能放的位置** p。什么叫“能放的位置”呢？就是对于所有不包含已经放过的位置的区间，如果它包含 p，则它的 b 必须等于 p。

把 i 放进 p 后，我们会删除这些包含 p 的区间，因此会分裂出 i 左侧的和 i 右侧的两个“极长空连续段”。

这些极长段看着就很像区间 DP。但是，因为我们放的是**最左侧**的位置，所以左边的最大值位置 q 必须和 p 同时出现在某个区间中。否则，把 i 放在 q 上也是合法的。这对 q 提出了一个下界限制。

于是设 $dp_{l,r,i}$ 表示区间 $[l,r]$ 中的最大值下标必须 $\geq i$ 时，这段区间的方案数。复杂度 $O(n^3)$。

容斥

不用多说了吧。
由于容斥的枚举量是指数级的（枚举固定合法的集合 S），大多数题目中容斥后的分析才是重点。而容斥一般不是主要难点。

[NOIP2024] 树的遍历

给定一棵 n 个点的树。我们定义一种以边为基础的遍历方式。定义两条边相邻当且仅当它们有公共端点。初始时，所有边都未被标记。进行如下过程：

1. 选择一条边 b 作为起始边，将它打上标记。
2. 假设当前访问边为 e，寻找任意一条与 e 相邻且未被标记的边 f，将 f 作为新的访问边打上此标记。然后再次进入第 2 步。
3. 如果与 e 相邻的边都被标记，如果 $e = b$ 则遍历结束，否则将 e 设为遍历 e 之前的上一条边，再次进入第 2 步。

显然，这样能够遍历所有边。我们构建一张新图，新图中的点与原图中的边一一对应，且每次进行第 2 步时，在新图中将 e 和 f 对应的点连边。显然新图也是一棵树。

给定原树上的若干条边（称作关键边），求以这些边为起始边时，可能得出的本质不同的新树个数。

$1 \le n \le 10^5$。

题解

难点在于多个起始边可能生成相同的新树。
容斥，枚举关键边的子集 S，求 S 中的边都能作为起始边的方案数。
分析这些起始边的性质，不难得出它们必然在同一条路径上。路径中间的点贡献为 $(deg-2)!$，其余点的贡献为 $(deg-1)!$。
于是对这条路径用树形 DP 就能得到答案。复杂度 $O(n)$。

[AGC035F] Two Histograms

有一个 $n \times m$ 的网格，一开始全为 0。
你需要在每行选一个前缀 $+1$，每列选一个前缀 $+1$。选择的前缀可能为空。
求可能得到多少种网格。
$1 \leq n, m \leq 5 \times 10^5$。

题解

不妨设第 i 行选取的前缀长度为 a_i，第 j 列选取的前缀长度为 b_j。
求操作方式数量是好求的，然而不同的操作方式可能产生同样的网格。例如，当 $a_i = j, b_j = i - 1$ 时，改为 $a_i = j - 1, b_j = i$ 就是另一种相同的方案。
可以证明，两个不存在 $a_i = j, b_j = i - 1$ 的操作方式不会生成相同的网格。证明只需考虑第一个 a 不同的行即可。
于是我们只要求不存在 $a_i = j, b_j = i - 1$ 的方案数即可。考虑容斥（这里实际上是二项式反演），不难 $O(n + m)$ 算出答案。

[UNR #7] 反重：求熵

有 n 个整数变量 $x_1, x_2, \cdots, x_n$，值域为 $[0, T]$。
给定 m 条约束，第 i 条约束形如 $x_{u_i} - x_{v_i} \leq c_i$。
求给变量赋值的合法方案数。
$2 \leq n \leq 8$，$0 \leq m \leq 200$，$0 \leq T \leq 10^{12}$。

题解

(u_i, v_i) 相同的约束显然可以合并。接下来令 $c_{u,v}$ 表示 u 和 v 之间最严格的约束。

考虑最后一个变量 x_n。它会受到若干个限制，每个限制形如 $x_i - c_{i,n} \le x_n$ 或者 $x_n \le x_j + c_{n,j}$。

枚举这其中最严格的限制，不妨设下界的最严格限制为 $x_i - c_{i,n}$，上界的为 $x_j + c_{n,j}$。“最严格”这个条件会在 i, j 和其他变量之间加上一些限制，它们也是和题中限制的形式相同的。

因此，我们每次枚举 (i, j) 并删掉一个变量，总计枚举 $O\left((n!)^2\right)$ 次。枚举的同时要维护一个多元多项式。

这太慢了。注意到 x_n 的贡献形如 $(x_j + c_{n,j}) - (x_i - c_{i,n}) + 1$，我们可以将两部分分开计算（谁说这不是容斥呢），每次只枚举 i 和 j 中的一个。枚举量变为 $O(n!2^n)$，足以通过。

[NOI2021] 机器人游戏

有 m 个机器人，每个机器人分别对一条纸带进行操作。每条纸带分为 n 个格子，每个格子有 3 种状态：填 0，填 1 或者空着。每个机器人有个操作序列 S_i。机器人会被设定一个初始位置，然后如果纸带不是全空，就会依次按照 S_i 中的每个字符进行操作：

1. `R` 表示机器人向右走一格，如果右边是边界，机器人会爆炸；
2. `0` 表示如果机器人所在格子非空，则将该格子改为 0；
3. `1` 表示如果机器人所在格子非空，则将该格子改为 1；
4. `*` 表示如果机器人所在格子非空，则将该格子反转。

给定每个机器人对应的初始纸带状态 X_i，以及操作完后的纸带状态 Y_i。求一个位置 p，使得所有机器人以纸带的第 p 个格子为起始位置时，可以在不爆炸的情况下执行完所有操作，且操作完后的纸带状态满足要求。

这太简单了，因此你要求有多少组 X_i, Y_i 使得这个问题有解。

$1 \le n \le 32$，$1 \le m \le 1000$，$1 \le |S_i| \le 100$。

题解

先容斥，枚举合法的起始位置集合。那么以这些起始位置开始的机器人会在哪些格子上执行哪些修改是固定的。而且每个格子的修改独立，这就可以对每个格子计数，最后乘起来即可。这样是 $O(2^n nm)$ 的。

考虑优化。大致思路是，如果操作序列中有很多 R，那么要考虑起始位置不会多；否则，一个靠前的起始位置根本影响不到靠后的位置。

具体地，设操作序列中 R 有 c 个，则只有前 $n-c$ 个起始位置需要考虑。我们从前往后 DP，状态中只需记录前面的 c 个位置是否被固定为合法的起始位置。复杂度为 $O(2^{\min(c,n-c)}n^2m)=O(2^{\frac{n}{2}}nm)$。

最后用 bitset 优化就能除上一个 ω。

[NOI2023] 深搜

给定一棵 n 个顶点的树 T，以及 m 条两两不同的非树边。另外指定了恰好 k 个点是关键点。
求有多少种选择 m 条非树边的方案，使得将 T 和这些选择的边构成图 G 之后，存在某个关键点 x，使得 T 是 G 的一棵以 x 为根的 dfs 树。
$1 \leq k \leq n \leq 5 \times 10^5$，$1 \leq m \leq 5 \times 10^5$。

题解

首先有经典结论：能作为 dfs 树的充要条件是非树边只有返祖边。
因此，每条非树边会**覆盖**一些点，以这些点为根的时候这条非树边会成为横叉边，是不合法的。
容斥。枚举 S，表示必须合法的关键点集合。建出 S 的虚树，则合法的非树边有两种形态：

- 虚树外的某个子树中的返祖边；
- 返祖边两端在虚树的同一条边上。

设 f_x 表示以 x 为虚树的根时，只考虑子树内，得到的贡献。算出 f_x 后，枚举 x 并算出 x 子树外的非树边个数 cnt，然后用 $f_x \times 2^{cnt}$ 贡献答案。
接下来考虑 f_x 的转移。

题解

设 x 在虚树上的儿子集合为 P。$\prod_{y\in P} f_y$ 肯定是贡献的一部分，剩余部分有三类非树边：

- 对每个 $y \in P$，y 到 x 的路径上的非树边；
- 对每个 $y \in P$，y 到 x 的路径上的点的每个不在此路径上的儿子子树内的非树边；
- x 的不在虚树上的儿子子树内的非树边。

这些贡献对每个 x 的儿子独立。如果某个儿子子树内不选点，贡献很简单。

如果选点，则需要在 dfs 的同时，用数据结构维护每个 y 到当前根的路径上的非树边的贡献。这里要维护区间乘，上线段树即可。

然后合并所有儿子就完了……？

题解

其实，有一件事是不独立的：x 要在虚树上。如果 x 的儿子中只有 < 2 个的子树内有选点，则必须要求 $x \in S$。否则 $x \in S$ 或者 $x \notin S$ 都行。

当然这不是问题，DP 时记录选了 0 个，1 个还是 ≥ 2 个即可。

问题在于，如果 $x \notin S$ 且 x 只有恰好两个儿子子树内选了点（设为 p 和 q），则那些一段在 p 到 x 的路径上，一段在 q 到 x 的路径上的非树边没被算进去！

但这其实也没关系。这些边的贡献形如矩形 $\times 2$，全局求和。用扫描线可以轻松维护。

要减去错误答案，因此前面的 DP 要把选了 2 个和选了 ≥ 3 个分开。

总复杂度 $O((n+m)\log n)$。

结语

谢谢大家。