

计数

kradcigam

江苏省常州高级中学

2026.1.26

Contents

- [1 闲话](#)
 - [2 树形结构](#)
 - [3 区间结构](#)
 - [4 数位结构](#)
 - [5 容斥](#)
 - [6 杂题](#)
 - [7 谢谢大家](#)

Contents

1 闲话

2 树形结构

3 区间结构

4 数位结构

5 容斥

6 杂题

7 谢谢大家

闲话

有没有 jiazhichen844 大神!!!

Contents

1 闲话

2 树形结构

3 区间结构

4 数位结构

5 容斥

6 杂题

7 谢谢大家

引入

讲点 JSOI 特别喜欢考的东西，不过大概已经是时代的眼泪了。⊗。

潜入行动

题目描述

有一棵 n 个点的树，你需要在树上选恰好 k 个点，使得每个点都至少有一个邻居被选择。
求方案数对 998244353 取模。

数据范围

$$1 \leq n \leq 10^5, \quad 1 \leq k \leq \min\{n, 100\}.$$

做法

设状态为 $dp_{i,j,x \in \{0,1\}, y \in \{0,1\}}$, 其中 x 表示是否有儿子选了, y 表示结点 i 是否选了。

这样每次合并两个子树大小分别为 sz_a, sz_b 的子树，时间复杂度是 $O(\min\{sz_a, k\} \times \min\{sz_b, k\})$ 的。

看起来做了 $O(n)$ 遍，两个 $O(k)$ 项的卷积。

分析时间复杂度：方法 1

不妨假设 $sz_a < sz_b$, 我们分类讨论:

- 1 $sz_a < k, sz_b < k$: 考虑对 a 子树内的点算 sz_b 的贡献。每个点的 sz_b 贡献之和不超过 $2k$;
 - 2 $sz_a < k, sz_b \geq k$: 考虑对 a 子树内的点算 sz_b 的贡献。每个点最多只会作为 a 子树贡献 1 次;
 - 3 $sz_a \geq k, sz_b \geq k$: 这样的合并不超过 $O(\frac{n}{k})$ 次，每次时间复杂度 $O(k^2)$ 。

三种情况时间复杂度都是 $O(nk)$ 。

分析时间复杂度：方法 2

我们用 DFS 序分析，注意到一个子树的 DFS 序区间，而 $\min\{sz_a, k\} \times \min\{sz_b, k\}$ 相当于跨过两个 DFS 序区间的长度不超过 k 的区间数量。

注意到所有长度不超过 k 的区间数量是 $O(nk)$ 的，并且每个区间只会被算一次，在 DFS 序区间对应的点的 LCA 处，这显然是唯一确定的。

时间复杂度 $O(nk)$ 。

试一试！

现在你已经完全掌握了树形 DP！

棋无常树

题目描述

有一棵 n 个点的有根树，根节点为 1，第 i ($i > 1$) 个结点的父亲为 fa_i 。其中第 i 个结点有权值 a_i 。若 $a_i = -1$ 表示结点 i 的权值还未确定，可以在 $[0, n]$ 中的所有整数中任选，否则结点 i 的权值为 a_i 。

第 i 个点的子树有限制 b_i 。若 $b_i = -1$ 表示结点 i 的子树没有限制，否则令结点 i 子树内点权可重集合为 S_i ，则要求 $\text{mex}(S_i) = b_i$ 。求对 $a_i = -1$ 的赋权方案数，对 $10^9 + 7$ 取模。

数据范围

$$1 < n < 5000.$$

我的做法：状态设计

考慮用 DP 刻畫填數的过程。

我们观察到对于 i 子树，记子树内 b_i 的最大值为 mb_i ，则 $0 \sim mb_i - 1$ 的数都已经在子树内出现过，同时对于子树内的点 i ，若 $a_i \neq -1$ ，则 a_i 也确定出现过，于是我们记录一个数组 $vis_{i,j}$ 表示 i 子树内数 j 是否已经确定出现。

记 $f_{i,j,k \in \{0,1\}}$ 表示 i 子树内填的数中 $\neq mxb_i$ 且不为已经确定出现的数中，一共有 j 种不同的数。我们发现子树内有些点无法填 mxb_i ，于是我们需要记录 $k \in \{0, 1\}$ 表示 mxb_i 是否已经出现过。



合并子树

考虑如何处理子树合并的转移。

假设当前合并结点 x 及其儿子结点 y 的子树。我们先考虑对 vis 数组进行合并，若对于 i , $vis_{x,i} = \text{true}$ 且 $vis_{y,i} = \text{false}$ ，则我们对 f_y 做一遍 $O(sz_y)$ 的背包，表示钦定子树 y 内没有确定出现过的数中，是否有 i ; $vis_{x,i} = \text{false}$ 且 $vis_{y,i} = \text{true}$ 同理。

这样我们就将子树 x 和子树 y 内确定出现的数的影响消除了。

合并子树

我们考虑需要对未出现过的数进行合并。

对于 $f_{x,i}$ 的转移，我们从小到大枚举 i ，对于 $i \geq 1$ ，我们需要考虑 x 子树没有确定出现的第 i 种值，在 y 子树没有确定出现的值中是否出现过，于是我们可以每一轮对 y 子树内再做一遍 $O(sz_y)$ 的背包，表示钦定在子树 y 内的没有确定出现的数中，是否有子树 x 内第 i 种没有确定出现的值。

这样我们就可以不需要考虑两个集合相同元素的影响，直接进行卷积即可。

加入结点 x 的影响

合并完子树后，我们考虑加入结点 x 的影响。我们只需要对结点 x 值的情况，以及新的 b_x 限制进行讨论，并更新 vis_x 数组。

时间复杂度

我们分析一下时间复杂度：

- 对于合并 vis 数组，我们可以把 $O(sz_y)$ 视作给子树内每个结点增加 1 的势能；对于一个结点 x 其向上合并，每增加 1 的势能就说明其对应祖先中 vis 数组中 `true` 的个数增加 1，由于总共只有 $n+1$ 种值，因此这部分时间复杂度是 $O(n^2)$ 的；
 - 对于处理完 vis 数组的合并，合并 f 数组，我们可以把 $O(sz_x sz_y)$ 视作在 x 子树内选择一个结点，在 y 子树内选择一个结点，并对这一对点增加 1 的势能；对于一对结点 (p, q) ，只会在 $\text{LCA}(p, q)$ 处增加 1 的势能，由于总共只有 $O(n^2)$ 对点，因此这部分时间复杂度是 $O(n^2)$ 的；
 - 加入结点 x 时，容易做到 $O(sz_x)$ 的时间复杂度。

时间复杂度 $O(n^2)$ 。

另一个做法：不同的状态设计

首先特判掉 $b_i \neq -1$ 且 b_i 小于子树内某个 b_j 的无解情况。

考虑设 $f_{i,j,0/1}$ 表示 i 子树内大于的子树内的 $\max b$ 的点数为 j , 并且先不考虑这 j 个数怎么填的方案数, $0/1$ 表示 $\max b$ 是否出现了。

合并两棵子树时，不妨设它们的 $\max b$ 分别为 A, B ，且 $A \leq B$ 。如果 $A < B$ ，先考虑 $A+1 \sim B-1$ 在第一棵子树内的出现情况，因为最后有 0/1 的状态所以要特殊讨论 B 。

转移

为了方便起见，下面的转移都不会标注 f 的 0/1，具体的细节并不太复杂。

这个转移大概形如 $\binom{j}{k} (B - A - 1)^{j-k} f_{i,j} \rightarrow f_{i,k}$, 暴力做是 sz^2 的, 无法接受。

注意到把 f_i 看成多项式后这个操作相当于是带入 $x + B - A - 1$ ，所以维护点值可以做到 $O((B - A)sz)$ 。

变成点值形式后，合并两个子树时需要把点值补到 sz 之和个，这部分复杂度是 $O(sz_1 sz_2)$ 的，总复杂度不超过 $O(n^2)$ 。

转移

设严格在子树内的 $\max b = p$, $b_i = q$, 如果 $q = -1$, 合并完子树和根节点之后可以直接结束。

如果 $q = p$, 舍掉 $f_{*,*,1}$ 之后也直接结束。

如果 $q > p$, 就逐个考虑 $p \sim q$ 的出现, 如果子树存在 a_i 等于考虑的数, 那么操作是 $f(x) \leftarrow f(x + 1)$, 否则是

$f(x) \leftarrow f(x + 1) - f(x)$, 特殊考虑 p 和 q 。

这部分复杂度是 $O((q - p)sz)$ 的。

最后答案就是 1 节点的多项式额外考虑上 $\max b + 1 \sim n$ 的结果, 相当于查询一个点值。

时间复杂度

下面来分析一下做法的复杂度。

这个复杂度是，每个节点有一个不大于父亲的非负整数标记 A_i ，且 $A_1 = n$ ，然后在一个节点处会付出 $(A_i - \max A_{son})sz_i$ 的复杂度。

统计每个节点对复杂度的贡献，可以发现就是 $A_1 - A_i$ ，因此时间复杂度 $O(n^2)$ 。

I'm Here

题目描述

有一棵 n 个结点的有根树，根结点的编号为 1。保证 $1 \sim n$ 可以构成这棵树的一个 dfs 序。初始时，所有的结点都没有崩溃。

每一天，LS 会探索一个没有崩坏的结点 u 。在这次探索后， u 及子树中所有点崩坏。

同时，在第 i 天 LS 的探索结束后，第 $n - i + 1$ 号结点若没有崩坏，则会崩坏。

分别对 $i \in [1, n]$ 求 LS 有多少种恰好探索 i 天的探索方案，满足最后一次探索的是 1 号结点，对 998244353 取模。

数据范围

$$1 < n < 80.$$

闲话

赛时不会。 😐

解法

题目等价于对于每个 i 求选出 i 个点的排列的方案数，满足如下 2 个限制：

- 1 若点 u, v 都被选中且 u 是 v 的祖先，那么在排列中 u 在 v 后面。
 - 2 若点 u 被选中，那么 u 必须出现在序列中前 $n - i + 1$ 个位置。

只有限制 1 是内向树拓扑序计数，直接树上背包记 $g_{x,i}$ 表示 x 子树内选了 i 个点的方案数。对于限制 2，注意到序列只有后缀最大值需要考虑，其他位置只要满足限制 1 即可。考虑从大到小将每个被选中的数插入序列，如果插入时插在了序列末尾，那么其为后缀最大值。

设 $f_{i,j,k}$ 表示填完了 $\geq i$ 的数，填了 j 个，强制当前序列最后一个数填在位置 k 的方案数。

转移

分类讨论： i 被选中时：

- i 若为后缀最大值: i 填在当前序列末尾, 枚举一个位置 x , 满足 $k < x \leq n - i + 1$, 从 f_{i+1} 转移过来。显然满足限制 1。
 - i 若不为后缀最大值: 枚举子树内包括 i 共选了 x 个数, 那么这 x 个数填在 k 前面即可。有

$$f_{i,j+x,k} \leftarrow f_{i+sz_i,j,k} \times g_{i,x} \times \binom{k-j}{x}.$$

复杂度 $O(n^4)$ 。

Contents

- 1 闲话**
 - 2 树形结构**
 - 3 区间结构**
 - 4 数位结构**
 - 5 容斥**
 - 6 杂题**
 - 7 谢谢大家**

区间集合

题目描述

给定两个正整数 n, c , 定义一个集合 S 是好的当且仅当

$S \subseteq \{[l, r] \mid 1 \leq l \leq r \leq n, l, r \in \mathbb{N}\}$ 且对于任意满足

$[a, b], [x, y] \in S$ 且 $\max(a, x) \leq \min(b, y)$ 的两个区间 $[a, b], [x, y]$

都有 $[\max(a, x), \min(b, y)], [\min(a, x), \max(b, y)] \in S$ 。

求所有好的集合的 $c^{|S|}$ 之和对 $10^9 + 7$ 取模的结果。

数据范围

$$1 \leq n \leq 700, 1 \leq c < 10^9 + 7.$$

解法

可以发现最后 S 集合一定形如一些不交的极长区间以及它们的子区间。

因此考虑对每个 i 计算有一个极长区间 $[1, i]$ 的前提下 $n = i$ 的答案，最后背包合并一下即可。

转移考虑删掉所有右端点是 i 的区间，剩下的区间还是形如一些不交的极长区间和它们的子区间。

解法

考察一个极长区间 $[l, r]$, 假设存在 $[c, i] \in S$ 满足 $l \leq c \leq r$, 首先就需要有 $[c, r] \in S$ 。

而我们可以发现每个极长区间 $[l, r]$ 内部的方案其实是相对独立的：用 $[c, i]$ 和一个 $[l, r]$ 的子区间生成的交一定和 $[c, r]$ 和这个子区间生成的交一样，因此不需要考虑交生成的区间的合法性。这样的话只会再生成一些形如 $[d, i]$ 的区间。

进行一些讨论可以发现，设所有以 r 为右端点的区间为

$[a_1, r], [a_2, r], \dots, [a_k, r]$, 且 $a_1 < a_2 < \dots < a_k \leq r$, 那么所有满足 $a_i \leq c \leq r$ 的 $[c, i] \in S$ 构成的集合是 $[a_1, i], [a_2, i], \dots, [a_p, i]$, 即左端点是 a 序列的一个前缀, 同时可以发现这样总是合法的。

解法

把没被任何区间覆盖的 i 也处理一下，即可得到一个做法：

设 $f_{i,j}$ 表示有一个极长区间 $[1, i]$ 的前提下 $n = i$ 且恰有 j 个形如 $[x, i]$ 的区间时的答案，那么转移就是对 $f_{i,j}$ 的 j 维度做后缀和然后做二维背包，要特殊考虑最后一个区间。

直接做是 $O(n^4)$ 的，可以对 j 维度拉插优化到 $O(n^3)$ 。

能量分配

题目描述

有 n 颗不同的宝石和 m 位学生，对于所有宝石分配给学生的方案，计算权值和，对 317 取模：

记第 i 位学生拿到的宝石数量为 c_i , 满足 $\sum_{i=1}^n c_i = m$ 。第 i 位同学的满意度 a_i 初始为 1, 考虑其他所有同学 $j \neq i$:

- 若 $c_j > c_i$ 则 $a_i \leftarrow a_i \times A_{i,j}$;
 - 若 $c_j = c_i$ 则 $a_i \leftarrow a_i \times B_{i,j}$ 。

一个方案的权值为 $\prod_{i=1}^n a_i$ 。

有 q 次询问，每次给定 n 。

数据范围

$$1 \leq q \leq 100, \quad 1 \leq n \leq 10^{18}, \quad 1 \leq m \leq 12.$$

闲话

这个题确实还是太变态了。

有人赛时这道题做了 2.5h 没调出来，宇宙级战犯。

是谁呢？

小模数的性质

以下记 $P = 317$ 。

首先对于固定的 c_i 序列，其方案数为 $\frac{n!}{\prod c_i!}$ 。

Kummer 定理

p 在 $\binom{n+m}{n}$ 中的幂次等于等于 $n+m$ 在 p 进制表示下进位的次数。

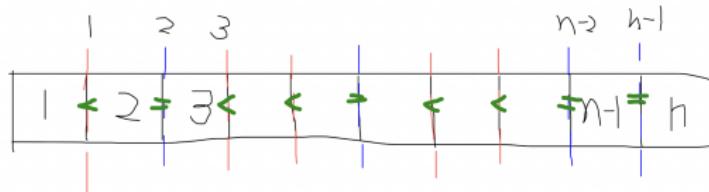
因此 $\frac{n!}{\prod c_i!} \not\equiv 0 \pmod{P}$ 当且仅当 $\sum c_i$ 做加法时没有产生进位。而此时，方案数在在 P 进制下每一位上是独立的。

解法

核心观察

考虑如何处理 A, B 权值的计算。首先我们发现，我们可以在最终排名（所有无标号同学之间的相对大小关系）确定下来后，再去关心学生和排名位置的对应关系，此时只需要进行形如将“人”填进排名的过程。

而“所有无标号同学之间的相对大小关系”可以用 $n - 1$ 个二进制位来表示：



解法

所以，我们的计算过程可以分成两个部分：

- 对于每个最终“所有无标号同学之间的相对大小关系”，计算对应的 A, B 部分权值和，即“将人填入排名的过程”；
 - 对于每次询问的 n ，求对应每个“所有无标号同学之间的相对大小关系”的方案数。

第一部分

这部分对时间复杂度的要求并不高，我们考虑一个暴力做法。

记 f_{S_1, S_2} 表示当前已经确定了选手集合 S_1 ，以及他们对应的“无标号相对大小关系”。转移时枚举 S'_1 满足 $S'_1 \cap S_1 = \emptyset$ ，即枚举“无标号相对大小关系”中在 S_1 前的一段相等的集合，并预处理 S_1, S'_1 之间的相对贡献。

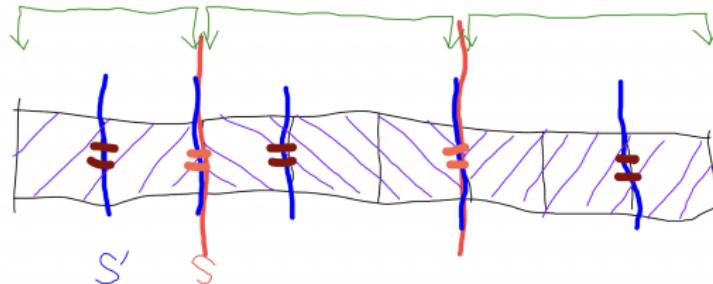
时间复杂度 $O(\sum_{i=1}^m \binom{m}{i} 2^i 2^{m-i}) = O(2^m \sum_{i=1}^m \binom{m}{i}) = O(4^m)$, 已经足够优秀。

第二部分

我们考虑先将 n 拆成 P 进制，假设有 k 位，第 i 位为 p_i 。

记状态为 $f_{i,S}$ 表示考虑完前 i 位 P 进制位，“所有无标号同学之间的相对大小关系”为 S 的方案数。考虑 $f_{i,S} \rightarrow f_{i,S'}$ 的转移 ($S \subseteq S'$)，对这部分预处理转移系数 g_{S,S',p_i} 。

考虑如何计算 $g_{S,S',i}$, 我们发现这个部分等价于对于 S 内的每个连续段先分别预处理 $h_{S,j}$, 再对 i 这一维做加法卷积。



计算 h

我们考虑一边搜索，一边记录 DP 数组 $dp_{i,j}$ 表示上一个数的值为 i , 总和为 j 的方案数。

根据 S 下一位的取值进行转移。

时间复杂度 $O(2^m P^2)$, 已经足够通过。

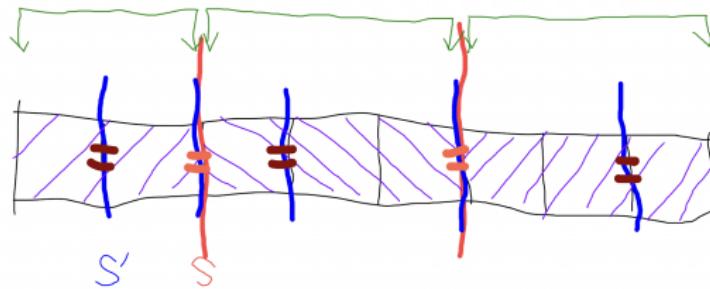
一个优化是，我们可以从大往小搜索，这样搜到第 x 个数时，

$j \geq ix$, 所以 $i \leq \frac{n}{x}$ 。

时间复杂度 $O(2^m P \log P)$ 。

计算 f

直接暴力枚举。时间复杂度 $O(3^m mP^2)$ ，无法通过。
但我们发现



绿色部分的顺序是不重要的，对这一维加入记忆化就足以通过。

fun fact

```
sort(all(g));
if (mp.count(g)) {
    auto gw = mp[g];
    F(t, 0, MOD - 1) w[tot][t] = gw[t];
} else {
    array<int, MOD> t1;
    t1.fill(0);
    t1[0] = 1;
    for (int t: g) {
        F(a, 0, MOD - 1) {
            int g = ww[t][a];
            if (g) {
                F(b, 0, MOD - 1 - a)
                    t2[a + b] += C[a + b][a] * t1[b] * g;
            }
        }
        F(a, 0, MOD - 1) {
            t1[a] = t2[a] % MOD;
            t2[a] = 0;
        }
    }
    mp[g] = t1;
    F(a, 0, MOD - 1) {
        w[tot][a] = t1[a];
        t1[a] = 0;
    }
}
```

这样就足以通过！

这么复杂的题目，对我这种老年退役菜鸡还是太不友好了!!! 😅

Contents

- 1 闲话**
 - 2 树形结构**
 - 3 区间结构**
 - 4 数位结构**
 - 5 容斥**
 - 6 杂题**
 - 7 谢谢大家**

Asterism Stream

题目描述

给定变量 x , 初始为 1, 每次等概率随机进行以下两种操作之一:

- 1** $x \leftarrow x + 1;$
 - 2** $x \leftarrow x \times 2.$

求期望多少次操作之后 x 会 $\geq n$, 对 998244353 取模。

题目描述

T 组数据, $T \leq 100, n \leq 10^{18}$ 。

基于数位 DP 的做法

令 $n := n - 1$, 根据期望线性性, 对所有 $x \leq n$ 求落在 x 的概率, 求和就是答案。

直接 $O(\log n)$ 枚举 $\times 2$ 的次数 L , 然后有额外系数 2^{-L} 。接下来考虑对非负整数向量 $(c_0, c_1, \dots, c_{L+1})$ 计数, 使得:

$$(c_0 + c_1) + c_2 \times 2 + c_3 \times 2^2 + \dots + c_{L+1} \times 2^L = n - 2^L.$$

其中系数贡献为 $2^{-(c_1+c_2+\dots+c_{L+1})}$, 注意此时和 c_0 无关, 因为 c_0 的存在是把 $\leq n$ 强制转成了 $= n$ 而设立的。

做法

我们把 c_i 也拆成二进制，则 2^i 的部分由 $c_0, c_1, \dots, c_{\min\{i+1, L+1\}}$ 这些系数确定。且他们都是属于 $\{0, 1\}$ 的。于是可以求出 $f_{i,j}$ 表示此时 $\sum c_j = j$ 的系数和。

求出 f 以后，我们令 $dp_{i,j}$ 表示考虑完了前 i 位且当前进位是 j 的方案数，然后直接枚举当前这一位 k 即可。

单次复杂度是 $O(\log^3 n)$, 由于要枚举 L 次, 于是总时间复杂度是 $O(\log^4 n)$ 。

时间复杂度 $O(T \log^4 n)$ 。

Bonus

上述做法能通过拉格朗日插值优化到时间复杂度 $O(T \log^3 n)$ 。

另一种完全不同的做法

令 $n := n - 1$, 我们令 f_i 表示经过 i 的概率, 于是答案为 $\sum_{i=1}^{n-1} f_i$ 。

对于 f , 有转移:

$$f_i = \begin{cases} \frac{1}{2}(f_{i-1} + f_{\frac{i}{2}}) & 2 \mid i \\ \frac{1}{2}f_{i-1} & 2 \nmid i \end{cases}$$

令 $low(x)$ 表示 x 中最低位的 1 的所在位，即 `builtin_ctz`。

转移

经过分析可得：

$$f_i = \sum_{j=0}^{\text{low}(i)} \frac{1}{2^{j+1}} f_{\lfloor \frac{i-1}{2^j} \rfloor}$$

这里我们令 $f_0 = 2$, 这样 $f_1 = \frac{1}{2}f_0 = 1$ 。

于是, 我们维护向量 A , 其中 A_j 表示 $f_{\lfloor \frac{j}{2^j} \rfloor}$

转移

我们发现对于相同的 $low(x)$, 其转移矩阵是相同的。

于是，我们预处理矩阵乘法。由于只关心 $low(x)$ ，所以从高位往低位转移即可。

时间复杂度 $O(T \log^3 n)$ 。

Contents

1 闲话

2 树形结构

3 区间结构

4 数位结构

5 容斥

6 杂题

7 谢谢大家

闲话

我也不太会。

Be Careful 2

题目描述

$n \times m$ 的矩形内部有 k 个禁止点，第 i 个禁止点位于 (x_i, y_i) 。
请计算内部（边界上不算内部）不含有禁止点的正方形的总面积，对 998244353 取模。

数据范围

$2 \leq n, m \leq 10^9$, $1 \leq k \leq 5000$, 禁止点坐标互不相同。

解法

考虑容斥，枚举正方形内的禁止点集合 S ，令包含禁止点集合 S 的正方形数量为 $h(S)$ 。

于是答案为 $\sum_S (-1)^{|S|} h(S)$ 。

我们定义 $f(S)$ 为一个点集，进行下述过程生成：

- 若有点在 S 的最小覆盖矩形角上，则将其与角的两条边界匹配；
 - 否则再对剩下未选有点的边界，选择在这条边界上编号最小的点。

不难发现对于相同的最小覆盖矩形的 S , $f(S)$ 都是相同的。

解法

定义 $g(S)$ 表示最小的禁止点编号 x 满足, $x \notin f(S)$, 且 x 在 S 最小覆盖矩形内部 (边界上也算内部); 若不存在这样的 x , 则 $g(S) = n + 1$ 。

那么对于相同的最小覆盖矩形的 S , $g(f(S))$ 都是相同的。我们构造双射, 对于点集 S , 若 $g(f(S)) \neq n + 1$, 则:

- 若 $g(f(S)) \in S$, 则将 S 和 $S \setminus \{g(f(S))\}$ 匹配;
 - 若 $g(f(S)) \notin S$, 则将 S 和 $S + \{g(f(S))\}$ 匹配。

由于两者之间的最小覆盖矩形相同，所以 $g(f(S))$ 也是相同的，所以确实是一组双射。

解法

于是我们只需要对 $g(f(S)) = n + 1$ 的点集 S 进行计数。

不难发现这样的 S 仅有 n^2 种，于是直接枚举即可。

计算 $h(S)$ 是一个小常数次自然数幂和的形式，于是可以 $O(1)$ 计算。

时间复杂度 $O(n^2)$ 。

关于 f 的定义

其实只要能保证对于最小覆盖矩形相同的 S , $f(S)$ 相同即可。
不过这个的定义关心到后面如何找矩形的过程。

KD Tree

题目描述

有一个长为 n 的点列 $\{(i, p_i)\}$, 每次可以选择 x/y 和某个坐标将点列分成左右/上下两边 (保持两边相对顺序不变), 然后分别递归下去, 直到区间长度为 1。求可以得到多少本质不同的点列, 对 $10^9 + 7$ 取模。

数据范围

$1 \leq n \leq 30$, p 为长度为 n 的排列。

解法

我们考虑记状态为 $f_{l,r,u,d}$ 表示计算划分点集 $S = \{i \mid l \leq i \leq r, u \leq p_i \leq d\}$ 的方案数。

注意到由于同一个点集排列可能有多种划分方式，则我们的思路是考虑计算字典序最小的划分方式。

此时，如何钦定操作的字典序就非常关键，这道题里的排列方式为：

- 我们将这个点集的横纵坐标离散化，假设有 $k + 1$ 个点，则有横纵坐标各有 k 个操作，为 $x_1 < x_2 < \dots < x_k$ 和 $y_1 < y_2 < \dots < y_k$ ；
 - 我们按照这样的顺序排列： $x_1, y_1, x_2, y_2, \dots, x_k, y_k$ 。

解法

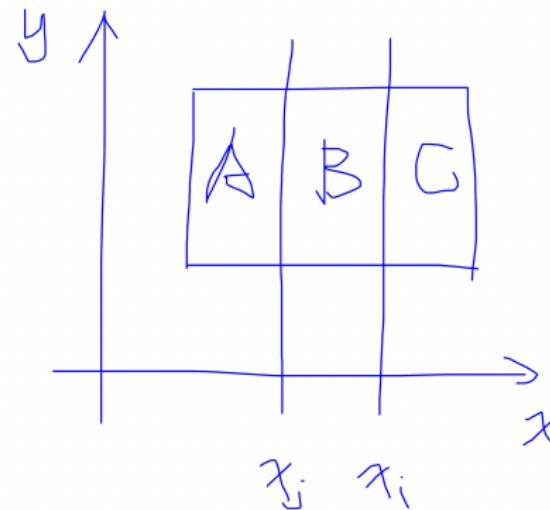
我们记状态：

- fx_i 表示第一步操作 x_i , 对于 $f_{l,x_i,u,d}$, x_i 是字典序最小的第一次操作的方案数;
 - fy_i 表示第一步操作 y_i , 对于 f_{l,r,u,p_i} , y_i 是字典序最小的第一次操作的方案数。

解法

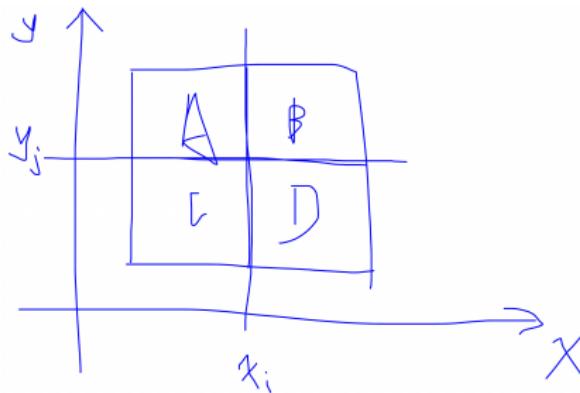
我们考虑转移：

- 首先假设当前计算字典序最小的操作是 x_i , 那么我们考虑容斥, 减去第一步字典序更小的方案:
 - 假设第一步字典序更小的操作为 $x_j(j < i)$, 那么此时:



解法

- 假设第一步字典序更小的操作为 y_j ($j < i$)，那么此时：



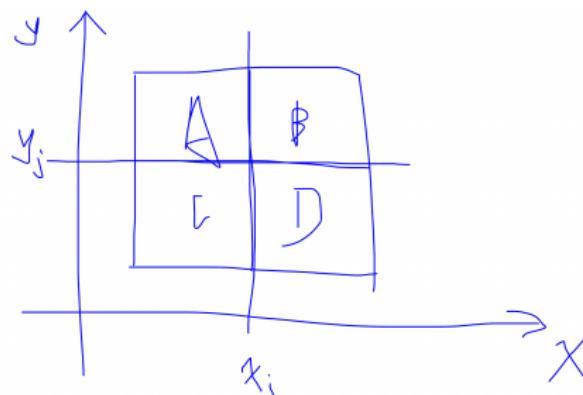
- 先 x_i : (AC) (BD);
 - 先 y_j : (CD) (AB)。

注意到 $j < i$, C 部分不可能塞 i 个点, 所以 A 部分必然不可能为空, 于是如果存在方案, 则必须要求 D 为空。

方案为 CAB: $f y_j f_l, x_i, y_{j+1}, d f_{x_i+1, r, y_{j+1}, d} \circ$

解法

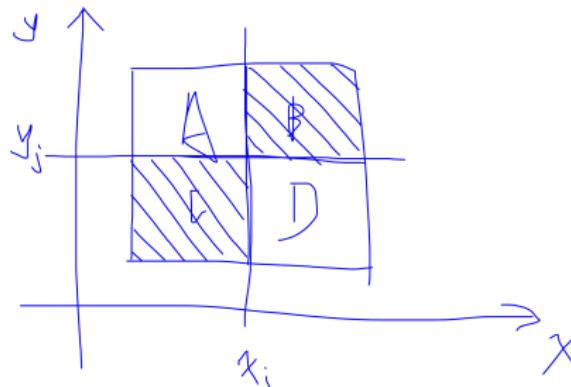
- 对于计算字典序最小的操作是 y_j , 唯一的区别是:
 - 假设第一步字典序更小的操作为 $x_i (i \leq j)$, 那么此时:



- 先 x_i : (AC) (BD);
- 先 y_j : (CD) (AB)。

解法

- 由于可能存在 $i = j$, 所以 A 部分有可能为空, 但注意到此时 $\{t | p_t \leq y_j\}$ 的点一定全放在了 C 部分, 所以 D 部分为空。



不难发现, 此时令空点集方案数为 1, 按照之前的计算方式要求 D 部分为空, 仍然是正确的。

时间复杂度 $O(n^6)$ 。

比赛

题目描述

给定长度为 n 的环，把 $1 \sim n$ 填入这个环，每个元素有一个颜色，如果相邻两个元素异色，那么编号较大的元素所属的颜色得分 ± 1 。

特别的，如果某个蓝色元素顺时针下一个元素是红色的，且红色元素权值较大，则这种情况不会让红色得分 +1。

对于 $k \in [-n, n]$, 计数有多少个有圆排列, 使得红色得分比蓝色得分多 k 。

数据范围

$$n < 3000.$$

闲话

错完了，这场比赛做了比赛不仅这场比赛爆了，之后也不会正常打比赛了。

这里只讲 85 分对应的 $n \leq 500$ 。

世界毁灭了，怎么 $\Theta(n^4)$ 卡常是可以获得这部分分数的。而我赛时简单尝试卡常无果之后就放弃了，写拆贡献的做法又一直在假。

解法

令环中 4 种相邻异色大小关系出现情况的次数分别为

$C_{B < R}, C_{B > R}, C_{R < B}, C_{R > B}$, 那么原题要求的就是

$C_{R>B} - C_{B>R} - C_{R<B}$, 看起来就非常奇怪且非常刻意, 这个不对称的东西实际上提醒我们需要拆贡献。

由于 RB 数量和 BR 数量一样，所以

$$C_{B \leq R} + C_{B \geq R} = C_{B \leq B} + C_{B \geq B},$$

$C_{B \geq B} = C_{B \leq B} + C_{B > B} - C_{B \leq B}$, 因此原题要求的便是

$C_{B \leq R} - 2C_{R \leq B}$, 就只有 2 个量了, 并且还都是 $<$ 。

我们考虑容斥，计算 $f_{x,y}$ 表示钦定有 x 个 $B < R$ 有 y 个 $R < B$ 的方案数，然后再通过对两维分别二项式反演就能求出 $g_{x,y}$ 表示恰好有 x 个 $B < R$ 有 y 个 $R < B$ 的方案数，进而计算出答案。

求解 f

由于 1 前的位置一定不会被钦定，考虑将 1 的位置放在开头，断环成链。我们发现钦定一组大小关系实际上相当于连一条边，这 $x + y$ 条边会形成 $n - x - y$ 条链。由于这是一张 RB 两种颜色的二分图， $R < B$ 是 R 的出边、 B 的入边； $B < R$ 是 B 的出边、 R 的入边，因此这两部分是独立的。

我们记连 x 条 $B < R$ 边的方案数为 p_x , 连 y 条 $R < B$ 边的方案数为 q_y 。两者都可以通过倒着 DP 求出。

那么, $f_{x,y} = p_x q_y (n - x - y - 1)!$, 后面的阶乘是因为链之间的顺序可以随意排列。

时间复杂度 $O(n^3)$, 二项式反演可以用 NTT 优化卷积做到 $O(n^2 \log n)$ 。

也许是更为高明的比赛策略

然后决定先做做 A，发现做连续段 DP 确实可以多项式复杂度，但感觉要记很多个元。考虑到做 CF2018F3 时太急着写记太多元的暴力导致简单做法没看出来的惨痛经历，我想想觉得肯定有什么简单转化，还是别往记一堆元的暴力上面想了。因为 $n \leq 500$ 的包有足足 45 分，觉得目标应该是一个 $O(n^3)$ 做法。拆了会贡献之后貌似确实想出来一个，然后就试着写了一下，调着调着发现假了！此时已经过了 1h，我意识到我又犯错了：应该先写一个 $O(n!)$ 暴力确认贡献是否拆对的。感觉人有点红温，决定先换到 B 做一做。

B 感觉做法很显然啊！把这个过程画出来并倒着做，一下就看出来怎么做了！但一开始细节没全编对，所以还是对拍调了一会，还剩 3h 时才过。

回过头来看 A，决定继续找简单转化。在试了很多很多种拆贡献的方法之后我终于发现了一个重要事实：按照顺时针转环，RB 的出现次数总和 BR 相等，而可以拆贡献使得这两种情形的权值分别是 1 和 -1！基于此又想了想终于得到了一个 $O(n^3)$ 做法，赶紧写了一下在还剩 1.5h 的时候拿到了 85 分。显然可以 FFT 优化到 $O(n^2 \log n)$ ，但最后要用 FFT 做的问题形式略有点特殊，说不定会有更简单的做法从而标算不是这个。当时没有意识到虽然 $n = 3000$ 但其实常数可以分析出是比较小的，同时我只会背效率不太优秀的递归式 FFT 的板子，再考虑到最后一个包只有 15 分，决定还是先不写了，毕竟我 C 现在还是 0 分。

Contents

- 1 闲话**
 - 2 树形结构**
 - 3 区间结构**
 - 4 数位结构**
 - 5 容斥**
 - 6 杂题**
 - 7 谢谢大家**

Random Variables

题目描述

对于一个长度为 n 取值为 $[1, m]$ 的正整数数组 a , 求其中出现次数最多的数字的出现次数的期望。

数据范围

$$1 \leq n \leq 10^3, \quad 1 \leq m \leq 10^9.$$

解法

考虑枚举 k , 计算 f_k 表示所有数出现次数不超过 k 的方案数。记 $dp_{i,j}$ 表示在所有数出现次数不超过 k 的前提下, 有 i 个数, j 种取值的方案数, 即 $f_k = dp_{n,m}$ 。

转移

考虑容斥计算 $dp_{i,j}$, 我们让第 i 个数随便填, 方案数为 $j \times dp_{i-1,j}$ 。

考虑减去不合法方案数，那么第 i 种数一定恰好出现了 $k+1$ 次，所以不合法方案数为 $j \times \binom{n-1}{k} \times dp_{i-(k+1), j-1}$ 。

由于 j 每减小 1, i 就会必定减小 $k+1$, 所以 $m-j$ 只有 $\frac{n}{k+1}$ 种取值。

于是暴力 DP 即可。

时间复杂度 $O(n^2 \log n)$ 。

Imbalance

题目描述

定义一个 01 串是平衡的当且仅当它的 01 个数相同。

现在给定 n, k 和一个长度为 m 的 01 串 S , 求有多少长度为 n 且以 S 为前缀的 01 串满足其不存在长度为 k 的平衡子串, 对 998244353 取模。

数据范围

$$1 \leq m+1 \leq k \leq n \leq 114.$$

解法

首先，这个字符串一定要么一直满足 1 的个数比 0 的个数多，要么一直满足 0 的个数比 1 的个数多，我们对两种对称的情况分别考虑即可。我们现在考虑 1 的个数比 0 的个数多的情况。

对于 $k \leq 22$, 我们可以跑状压的暴力。

对于 $k > 22$, 我们考虑将整个字符串按 k 分成若干行, 于是最多只有 5 行。

解法

暴力

直接 DP 复杂度，复杂度是 $O(n^9)$ ，无法通过。

定义前缀值 s_i 表示前 i 个字符中 1 的个数。

考虑我们的限制是 $s_i - s_{i-k} > \frac{k}{2}$ 也就是 $s_i > s_{i-k} + \frac{k}{2}$ 。

解法

我们考虑将第 i 行的折线向下平移 $\frac{ik}{2}$ ，于是限制变成了

$$s_i > s_{i-k}.$$

将 s_i 画在 $(i \bmod k, s_i - \lfloor \frac{i}{k} \rfloor \frac{k}{2})$ 。

于是，合法路径等价于，折线两两不交，这是 LGV 引理能做的事情。

所欲我们先枚举每行终点前缀和的值，而确定字符串的前缀，等价于把起点平移。

时间复杂度 $O((\frac{n}{k})^3(\frac{k}{2})^{\frac{n}{k}})$ 。

Minimums or Medians

题目描述

你有一个包含 $1 \sim 2n$ 共 $2n$ 个整数的集合 S 。你必须执行恰好 k 次操作，每个操作都是以下两种其中之一：

- 将 S 中第 1, 2 个元素删去。
 - 将 S 中第 $\frac{|S|}{2}, \frac{|S|}{2} + 1$ 个元素删去。（显然 $|S|$ 一直是偶数，所以 $\frac{|S|}{2}$ 也一定是整数）

请你统计，通过这些操作可以获得多少个本质不同的最终集合 S ？答案对 998244353 取模。

数据范围

$$1 < k < n < 10^6.$$

解法

我们考虑找充要条件：

- 1 注意到所有被删除的连续段长度都是偶数。并且不同的连续段之间，都是被分开删除的。
注意到只有从 1 开始的连续段才可能用操作 1 删除，于是其它被删的数都是通过操作 2 删除的。

2 注意到第 i 次若为操作 2，则删除的较大量为 $i + n$ ，所以，被删除的数都必须 $\leq n + k$ 。

3 注意到若删除了数 $t \leq n$ 不在第一段，则 $[t, 2n - t + 1]$ 的数都必须被删除。

解法

上述 3 个必要条件是充分的，因为考虑我可以通过构造，每次能用操作 2 就用操作 2，否则就用操作 1，不难发现一定是合法的。设函数 $f(n, m)$ 表示长度为 n 的段中，选出 $2m$ 个数，并要求选出的数形成长度为偶数的连续段。考虑捆绑法，我们没选出一个数就要求必须选它后面的那个数，于是这样就只有 $n - m$ 个物品，从中选出 m 个，所以是 $\binom{n-m}{m}$ 。

解法

考慮如何計數。

首先，我们枚举从 1 开始的连续段需要操作的次数 t ，因为涉及到条件 3，所以我们需要根据是否覆盖超过一半来分类讨论：

- 1 $2t < n$, 于是再枚举在 n 之前被删掉的连续段长度 c , 于是得到:

$$\sum_{t=1}^{\min\{k, \lfloor \frac{n-1}{2} \rfloor\}} \sum_{c=0}^{\min\{k-\text{len}, n-2k-1\}} f(k-t, k-t-c)$$

注意到 len 增加时，组合数的变化可以 $O(1)$ 维护，类似莫队移动指针，加入或删除组合数。

解法

2 $2t \geq n$, 那么剩下的就可以随便操作, 于是对答案的贡献为:

$$\sum_{t=\lceil \frac{n}{2} \rceil}^k f(n+k-2t-1, k-t)$$

时间复杂度 $O(n)$ 。

Contents

- 1 闲话**
 - 2 树形结构**
 - 3 区间结构**
 - 4 数位结构**
 - 5 容斥**
 - 6 杂题**
 - 7 谢谢大家**

谢谢大家

祝大家在联合省选 2026 中取得好成绩！

Thanks!