袁桢淏 集训队试题准备

T1 CF575A Fibonotci

题目大意

有一个循环节长度为 n 的无限循环序列 s（从 0 开始编号），给定其循环节，并修改其中 m 个位置的值。定义 F 的递推关系式：

$$F_n = \begin{cases} 0 & (n=0) \\ 1 & (n=1) \\ s_{n-1}F_{n-1} + s_{n-2}F_{n-2} & (n \geq 2) \end{cases}$$

求 $F_k \bmod p$。

数据范围

- $n, m \in [1, 50000]$
- $k \in [0, 10^{18}]$
- $p \in [1, 10^9]$

解题过程

看到 $k \leq 10^{18}$ 和 “递推关系” 不难想到矩阵快速幂。

先不考虑 $m = 0$ 也就是不修改 s 序列的情况。先列出一般性的矩阵乘法表达式，考虑一段循环节内的转移矩阵可以合并成一个矩阵 S，再根据从 0 到 k 经过多少个循环节对 S 做快速幂，最后不足一个循环节可以暴力转移。

接下来考虑对其中 m 个位置进行修改。对于其中没有修改的循环节，仍可以直接使用矩阵快速幂进行计算；对于有修改的循环节，可以发现修改的位置将这个循环节分割成若干段，可以考虑提出每一段的矩阵操作后暴力处理修改位置的转移。

则只需要考虑快速求出一段区间 $[l, r]$ 的转移矩阵。可使用线段树维护。

代码细节较多。时间复杂度 $O(n + m(\log n + \log p))$。

T2 CF639E Bear and Paradox

题目大意

一场考试共 n 道题，每道题有一个初始得分 p_i 和一个解题时间 t_i，若此题在第 x 分钟解决，则它的得分为 $p_i(1 - c \cdot \frac{x}{T})$，其中 $T = \sum_{i=1}^{n} t_i$，$c \in [0, 1]$ 为你指定的一个“降分常数”。

在所有 $n!$ 种解题顺序中，存在至少一个最优的解题顺序使得其得分最高。

在某种解题顺序下，定义“悖”存在当且仅当存在一对 $i \neq j$， $p_i < p_j$ 且在第 i 题上的得分严格大于第 j 题。

现在问，在所有最优解题顺序下都不出现 “悖” 的情况下，c 最大为多少？

数据范围

- $n \in [2, 150\ 000]$
- $p_i \in [1, 10^8]$
- $t_i \in [1, 10^8]$

解题过程

题意较为复杂，剥离外壳考虑核心。对于制定的 c，考虑其最优解题顺序，凭直觉可以贪心，于是对于任意一种解题顺序，考虑相邻两个题，考虑它们之间交换后得分更高的条件（设其为 i, j，解到这两题时已经过去了 t 分钟）：

$$p_i - \frac{c(t+t_i)}{T} + p_j - \frac{c(t+t_i+t_j)}{T} < p_i - \frac{c(t+t_i+t_j)}{T} + p_j - \frac{c(t+t_2)}{T}$$
$$\Leftrightarrow t_j p_i < t_i p_j$$
$$\Leftrightarrow \frac{p_i}{t_i} < \frac{p_j}{t_j}$$

据此不难发现最优解题顺序一定是按照 $\frac{p_i}{t_i}$ 递增的顺序解题的。

若所有的 $\frac{p_i}{t_i}$ 都不同，则只存在唯一一个解题顺序，可以简单判断“悖”是否存在。

若有多个 $\frac{p_i}{t_i}$ 不同，考虑每道题的最早解决时间和最晚解决时间，可以证明这样不会对“悖”的存在性产生影响。

则考虑最大的 c，只需二分答案即可。

将对 p 排序的过程提到二分答案的外边，总复杂度为 $O(n \log n)$。

在编写代码时需要注意多个顺序需要分层处理（多个 p_i 相同的题目之间不算“悖”）。

T3 agc-039F Min Product Sum

题目大意

对于一个矩阵 $\{A_{ij}\}$，设 $v_{ij} = \min\{\min_{1\le k\le m} A_{i,k}, \min_{1\le k\le n} A_{k,j}\}$，则称这个矩阵的权值为 $\prod_{1\le i\le n}\prod_{1\le j\le m} v_{i,j}$。

现将矩阵中每个位置填上 $[1, K]$ 间的整数，问在所有 K^{nm} 种矩阵的权值和模 D。

数据范围

- $1 \le n, m, K, \le 100$
- $10^8 \le D \le 10^9$ 且为质数

解题过程

> 一般来说，一个矩形中如果格点与格点间没有差异的话，一般会考虑对行和列进行操作以降低计算量。

在这题中，先考虑可以把问题等价于成下面这个问题：

> 设存在两个 $n \times m$ 的矩阵 A, B（且每个格点的值都为 $[1, k]$），对于任意 i, j，有 $A_{i,j}$ 小于等于矩阵 B 中第 i 行与第 j 列的最小值。
>
> 求有多少对符合上述描述的矩形 A, B。

先考虑给 B 的每一行/列赋上一个最小值，使得这一行/列中每一个元素都不小于这个最小值。则对于任意一个位置 (i,j)，考虑其行列最小值分别为 a,b:

- 则其可在 $[\max\{a,b\},K]$ 上自由取值（权值）
- 且在矩阵 A 的对应位置中，这个位置的贡献为 $\min\{a,b\}$（方案数）
- 综合“权值”与“方案数”，就可以迅速算出符合设定“最小值”的矩阵的答案

但问题在于现在不知道每一行/列的最小值，结合观察到的数据范围，可以大胆使用动态规划解决此问题：设 $f[t][i][j]$ 表示已经设定完所有最小值小于等于于 t 的行和列了，且目前已经设定完了 i 行 j 列的“临时答案”（已经考虑了 i 行 j 列个格子的权值与方案数、i 行 $m-j$ 列个格子的权值、$n-i$ 行 j 列个格子的权值的答案）。转移则考虑对于每一个权值 t，都考虑有多少行/列的最小值恰为 t……

但是这个做法有一个问题，就是如此一来无法保证每一行每一列的实际最小值恰为设定值（每一个取值都大于等于最小值），所以可以在刚刚的做法中加入容斥。即对于每一个权值 t，额外考虑有多少行/列会“被容斥”。

同时，可以发现在每一层 t 转移中，以上四维可以分开考虑。

总时间复杂度 $O(Knm(n+m))$，空间复杂度 $O(nm)$。

时间限制可能有些紧，但考虑将一些共同部分进行一定程度的预处理可以通过此题。