省选 2024 模拟赛 Day 2 (2.17) 题解

By YeahPotato

2024.2.17

T1

算法 1

$\mathrm{O}(n^4)$ 直接二维前缀和。

$\mathrm{O}(n^3)$ 可以枚举上下边界，然后 `two pointers`。

算法 2

补集转化。枚举上下边界。列分为三类：

1. 全 0。
2. 一个 1。
3. ≥ 2 个 1。

连续全 0 段可以选，一个 1 列的两侧连续全 0 段可以选。

如果是逐渐将边界缩小，则可以用链表。由于每列只会在两行出现变化，故总时间复杂度为 $\mathrm{O}(n^2)$。

T2

算法 1

直接暴搜并打表出 $n=m=5$ 的答案，可通过子任务 1。

算法 2

对于 $n=3$，岩浆湖只可能出现在中间一行，可能可以进行一些较简单的讨论得到一个 `dp`，可通过子任务 2。
高斯消元或 `BM` 发现 $n=3$ 时答案关于 m 有一个 14 阶的递推式，可通过子任务 3。

算法 3

记陆地为 1，其余为 0，岩浆湖即为局部的 0 连通块。

核心的 `dp` 思路是记录当前轮廓线边缘的 1 的连通情况，这可以抽象成有编号的球放入无序的盒子中，特殊地，不会出现形如等价类为 $a0b0a0b$ 的交叉情况，且出现相邻 1 也可以剪枝。总之如果从组合意义分析状态结构会较复杂，不如直接记下 `dp` 值非零的所有状态。

转移的方式有整行和逐格转移，后者更快。

对于岩浆湖的计数，可以同样记录等价类，这样状态数会过多；可以通过 1 的等价类情况推导出数量，分析较复杂；考虑欧拉定理：将 1 视作点，八连通范围的 1 对之间连边，这样，面数 $F=1-V+E$，岩浆湖数量 $c=1-V+E-\#K_3+\#K_4$（K_4 破坏平面图的情况可以视作只连其中一条斜边，另一条再单独算）。考虑当前的单格决策对 c 的影响（右下角为新决策格，决策时连上与左侧、上方、左上，以及左与上之间的边）：

```
00  01  01
01  10  11
```

这三种情况分别贡献 $-1,+1,+1$，其余情况可讨论得到贡献均为 0。因此只需在当前 $(i,j)(i<n)$ 与 $(i+1,j-1)$ 格的 01 相同时额外记录 $(i,j-1)$ 选了 0 还是 1 即可。

状态与整数之间的映射可以用 `trie` 树。

如果做到了这一步，那么时间复杂度为 $\mathrm{O}(n^2mS(n))$，其中 $S(n)$ 为状态数（实际会随着行号变化，但变化不大）。可以使用类似 `dp` 套 `dp` 的方法先将所有可能到达的状态与转移处理出来，再直接在有向图上转移。这样可以少一个 n。

为保证 1 形成单个连通块，需要判掉一些不合法的转移，然后对于 1 最后出现在每一行的情况求和。

想到上文中一部分优化可以通过一部分子任务。出题人没实现过，所以就多放了几档。

前景

出题人在 $n=12$ 时跑出的状态数为 729026（对于所有行求和），这是没跑 `DFA` 最小化的数量，不知跑最小化效果如何。

对于这类 01 矩阵连通块计数的问题，目前应该只有指数的方法，例如 A086265。

n 较小时根据 `dp` 形式可判断出存在阶数较小的常系数线性递推式，不过意义不大。

T3

算法 1

注意到这个问题本质上是模意义下的积木大赛/铺设道路。转化题意如下：

给定整数 $a_{1\cdots m} \in [0, n)$，每次可以将一个区间全体减一，减到 -1 的变成 $n-1$，要求全部变成 k。

这个模型是有可能使某个数减 $\geq n$ 次的，例如：

```
1 2 3 0 1 2 3 0 3 2 1 0 3 2 1 (n=4)
```

因此不能直接做。一些暴力例如暴搜操作或枚举每个数多减几倍 n 可以通过前 1 或 2 个子任务。

算法 2

令 $f_{i,j}$ 表示第 i 个数，减了 $a_i + j \cdot n$ 次的答案。

1. $a_i \geq a_{i+1}$。这时有可能 a_{i+1} 比 a_i 多减一倍 n，转移包括 $f_{i,j} \xrightarrow{a_i - a_{i+1}} f_{i+1,j}$ 与 $f_{i,j} \xrightarrow{0} f_{i+1,j+1}$。
2. $a_i < a_{i+1}$。这时有可能 a_{i+1} 比 a_i 少减一倍 n，转移包括 $f_{i,j} \xrightarrow{0} f_{i+1,j}$ 与 $f_{i,j} \xrightarrow{a_i + n - a_{i+1}} f_{i+1,j-1} (j > 0)$。

关于为什么不可能变化其他倍的 n，可以用调整法证明（一侧代价减少 n，另一侧代价增加 $\leq n$），或者也可以理解成其他部分斜率为 0 或 n，会被凸壳两侧吞掉。

如果没想到转移剪枝和 `slope trick` 则需要 $\mathrm{O}(n^4)$ 或 $\mathrm{O}(n^3)$（默认 n, m 同阶），可通过前 3 或 4 个子任务

现考虑 `slope trick`。可以归纳证明 $f_{i,*}$ 形成一个下凸壳，斜率从某个 $> -n$ 的值升到 0。扫描差分数组，用大根堆维护斜率的相反数，最终答案为 $f_{m+1,0}$，即堆中的和 $+a_m$。

1. 相当于与 $(0, a_i - a_{i+1}) - (1, 0)$ 作闵可夫斯基和，即往堆里 `push` 一个 $a_i - a_{i+1}$。
2. 相当于与 $(-1, a_i + n - a_{i+1}) - (0, 0)$ 作闵可夫斯基和，即往堆里 `push` 一个 $a_i + n - a_{i+1}$ 再 `pop` 一下。

暴力做就是 $\mathrm{O}(n^2 \log n)$。可通过前 5 个子任务。

注意到在目标值变化时，$a_i - a_{i+1} \bmod n$ 是不变的，只是 `push` 之后是否要紧随 `pop` 会变化。并且目标值全部轮下来的过程中，一个 i 至多切换两次。因此现在要支持的就是部分可追溯化堆的操作。关于可追溯化堆详见 `kczno1` 的 `19` 年集训队论文。这里也稍微讲一下。

考虑所有最后留在堆里的元素 S。如果操作序列的一个前缀做完后堆中的元素是 S 的子集，则称这个前缀为一个“桥”。理解接下来的操作可以想象将 `pop` 操作与 `pop` 掉的值对应的 `push` 操作连弧线，所有弧线交织的连通块之间就是桥。

如果现在加入了一个 `pop`，那它会 `pop` 掉前面某个 `push` 的值。而这个值可能之前在后续 `pop` 中被 `pop` 掉，这样后面那个 `pop` 就要 `pop` 一个较小的元素，就导致了一系列连锁变化。可以将连锁变化理解成这样：如果当前插入 `pop` 将一个后续被 `pop` 掉的值 `pop` 掉了，那么可以视作加入的是后续那个 `pop`。这样变换下去，直到第一个桥，只需将桥中的最大值 `pop` 掉即可。如果初始 `pop` 加入的位置不是桥，那么后面第一个桥末尾一定是个 `pop`，所以这个没问题。

如果现在删除了一个 `pop`，那它会将它 `pop` 掉的值放回来，但这个值可能会被后面的 `pop` `pop` 掉，同样可以理解成删除的是后面第一个 `pop` 掉的值 $\leq$ 当前放回值的 `pop`，这样连锁下去。而后面 `pop` 中 `pop` 掉的最小值是初始删除 `pop` 位置之前（不含本身）最后一个桥之后不在桥中的最小值。为什么不用担心删除的 `pop` 与上一个桥之前的那些 `pop` 影响呢？因为他们 `pop` 掉的值一定更大，否则应该再断出一个桥来。

这题可以用线段树做。维护 $\pm1,0$ 序列（如果当前 `push` 被 `pop` 掉则 $+1$，如果紧随 `pop` 则 -1）、前缀和最小值及最左最右位置、最大的在 S 中的值、最小的不在 S 中的值。加入 `pop` 时找 i（含本身）以后的桥，这一定存在；删除 `pop` 时找 i（不含本身）之前的桥（可能为 0）。

时间复杂度 $\mathrm{O}(n\log n)$，子任务 6 留给卡常。

前景

本题有性质 $ans_{i+1}\le ans_i+1$，不知是否有不用可追溯化堆的方法。

联合省选 `2023 D1T3` 和 `2023` 年互测 `R15T1` 都是该算法的例题。值得注意的是这两题的官方题解模型均形如“限制每个时刻堆的大小，如果超过则 `pop`，然后进行一些 `push` 的修改”，这与可追溯化堆的标准表述等价——只需将过程倒过来，将大小根堆互换即可。