第 48 届国际大学生程序设计竞赛亚洲区域赛沈阳站

李泽仁

华为云计算与网络创新Lab

2023 年 11 月 12 日

C. Swiss Stage

签到题，当$y = 2$时必须要进行Bo3，答案为$2(3 - x)$
否则只有最后一局需要Bo3，答案为$4 - x$

J. Graft and Transplant

由于树的节点数至少是 2，每个点的度数至少是 1，称度数为 1 的点为“叶子”。

可以观察到将和 u 相连的边（除了 $u-v$）都接到 v 上时：

- 如果 u 是叶子，那么没有边发生改变，此时操作后的树和操作前的树同构；
- 如果 v 是叶子，那么把相当于只交换了 u 和 v 的标号，此时操作后的树仍然和操作前的树同构；
- 否则 u 会成为新的叶子，v 的度数得变大，其他点的度数不发生改变，此时叶子数量恰好增加了 1，因此操作后的树必然和操作前的树不同构。

J. Graft and Transplant

因此每一步合法操作都会使叶子数恰好增加 1，叶子数达到最大值 $n-1$ （除了 $n=2$ 时是 n）时游戏结束，只需关注初始局面叶子数以及最终局面的叶子数的奇偶性。

E. Sheep Eat Wolves

记 $f_{i,j,0/1}$ 表从初始状态开始要达到与家在河不同侧的岸边有 i 只羊和 j 只狼且农夫在与家不同侧/同侧的岸边时的最少步数，以 $f_{X,Y,0} = 0$ 为初值进行BFS，每次枚举和农夫一起乘船的羊和狼的数量进行转移，最终结果为 $\min_{j=0}^{Y} f_{0,j,1}$，复杂度 $O(n^4)$。

K. Maximum Rating

对于给定的rating变化序列，先考虑最小化最高rating发生变化的次数，可以发现如果先一直掉分，最高rating一直都是 0，之后按照涨分幅度从小到大上分就能最小化次数，再考虑最大化次数，显然是先涨分再掉分，此时可以从最小化次数的方案每次将涨分幅度最大的移到最开始，变化次数就可以取到最小到最大次数之间的所有值。

K. Maximum Rating

现在要支持修改，可以发现求出最大的 k 使得非正数之和加上正数的前 k 小之和 ≤ 0 之后 $k+1$ 即为答案，使用权值线段树维护所有正数的升序排序得到的序列并在线段树上二分即可，离线询问并对所有正数做离散化之后的复杂度是 $O((n+q)\log n)$。

M. Outro: True Love Waits

显然 s 走到 t 等价于 0 走到 $s \oplus t$。
考虑两个二进制位一组，将点标号写成四进制。从 0 出发，前四步是 $(0)_4 \to (1)_4 \to (3)_4 \to (2)_4 \to (0)_4$。这是一个基本循环结构，再走一步 $(00)_4 \to (10)_4$ 后，接下来四步会使最低位再循环一次这个基本结构，再走一步 $(10)_4 \to (30)_4$，观察到次低位也在最低位的外层循环这个基本结构，次次低位以此类推。

那么第 k 次到达 0 的步数为 $s_k = 4^1 + 4^2 + \cdots + 4^{k-1}$。对于其他点，一定能至少到达一次且步数是能按位累加贡献的，能重复到达的次数上限取决于末尾 0 的个数，答案只需第一次到达的步数再加上 s_k。

B. Turning Permutation

考虑从前往后确定排列中每一位的取值，先以第一位 p_1 为例，首先算出当 $p_1 = 1$ 时可能的回转排列的数量（记为 c），如果 $c \geq k$，那么说明字典序第 k 大的排列的 p_1 就等于 1；否则如果 $c < k$，那么可知字典序第 k 大的排列的第一位一定比 1 大，所以这个时候让 k 减去 c，并接着枚举 $p_1 = 2$，依次类推，直到 $c \geq k$ 为止。特别地，如果从 1 到 n 都枚举完了，c 还是小于 k，就说明 k 是大于回转排列总数的，输出 -1 即可。然后之后的每一位都可以用类似的逐位试验法来求得。

B. Turning Permutation

基于上述过程，我们就把问题转化成了求解 $O(n^2)$ 次计数问题，即在确定了排列的前若干位的情况下的回转排列数。考虑动态规划，设 $f[i][j][0/1]$ 表示把 $1, 2, \cdots, i$ 加入了排列，i 是当前 i 个数中的第 j 位，并且初始方向是 1 在 2 的左边/右边的情况下的回转排列数。由于本题数据范围比较小，所以可以直接暴力转移，根据 i 的奇偶性和 0/1 状态，来确定 $i+1$ 应该是在 i 的左边还是右边，然后枚举合法的 j 并转移，注意检查 $i+1$ 是不是排列被确定的前若干位，是的话合法的 j 就只能是被确定的前若干位中在 $i+1$ 前面且小于 $i+1$ 的数的个数。

可知动态规划中状态数是 $O(n^2)$，转移可以暴力地 $O(n)$，或者用前缀和优化降至 $O(1)$，然后要求解 $O(n^2)$ 次这个计数问题，所以整个题的时间复杂度是 $O(n^5)$ 或者 $O(n^4)$ 的。

H. Line Graph Sequence

记图 G 的点数为 n，边数为 m，第 i 个点的度数为 d_i，那么有 $\sum_{i=1}^{n} d_i = 2m$，注意到 $(d_i - 1)(d_i - 2) \geq 0$，对 i 求和并整理可得 $L(G)$ 的点数 $\sum_{i=1} \frac{d_i(d_i-1)}{2} \geq \sum_{i=1}^{n}(d_i - 1) = 2m - n$，这说明线图序列的点数下凸，因此当 $L^{t+1}(G)$ 的点数（也就是 $L_t(G)$ 的边数）不少于 $L^t(G)$ 的点数时，$L^t(G)$ 的点数即为所求。

H. Line Graph Sequence

不难分析当所有连通块都是链、环或者 $K_{1,3}$ 的时候无限迭代的线图序列是收敛的：

- 链每次迭代会少一个点，直到为空；
- 环每次迭代不会发生变化；
- $K_{1,3}$ 迭代一次之后会变成三元环，以后都不会发生变化。

H. Line Graph Sequence

对于存在连通块导致迭代发散的情况：

- 如果存在一个点的度数至少是 4，不难验证 $K_{1,4}$ 迭代不到 10 次点数就会超过 10^5，那么包含 $K_{1,4}$ 作为子图时迭代次数不会更多；
- 否则一定存在点的度数是 3，且整个连通块不是 $K_{1,3}$，这需要在 $K_{1,3}$ 的基础上至少加一条边，不难验证各种情况下迭代不到 10 次点数就会超过 10^5，那么包含这些图作为子图时迭代次数不会更多。

因此一个可行的做法是先尝试直接迭代 10 次，如果某次迭代的边数不少于点数，这时点数即为答案，否则 10 次迭代之后一定是收敛的情况，并且此时每个连通块只能是链或者环，直接计算每个连通块在完成剩下迭代操作之后的点数即可。

D. Dark LaTeX vs. Light LaTeX

题意为从s和t中分别选一个子串p和q，要求拼接后$p+q$为平方串。
先讨论p比q长的情况，则需要满足p可表示为aba，q表示为b的形式。

我们n^2枚举p中两个a串的左断点l_1和l_2，可以发现可行的a的最大长度是$\text{lcp}(S[l_1:], S[l_2:])$。b的最大可行长度是$[i \text{ from } 1 \cdots |T|]$ $\max(S[:l_2], T[:i])$。因此只需要简单的使用n^2 dp来预处理出lcp和lcs即可进行答案统计。

另一种p比q短的情况同理，时间空间复杂度都为$O(n^2)$

I. Three Rectangles

如果存在一个小矩形和大矩形的大小相同的情况，此时另外两个小矩形可以在大矩形里任意放置。
否则一定要有一个小矩形覆盖相邻的两个角：

- 如果小矩形和大矩形高度相同，可以覆盖第一列或最后一列的两个角；
- 如果小矩形和大矩形宽度相同，可以覆盖第一行或最后一行的两个角。

I. Three Rectangles

这里可以考虑容斥，对每个小矩形钦定是否覆盖两个角以及覆盖哪两个角，考虑未被钦定的小矩形个数：

- 如果没有矩形未被钦定，需要判断三个小矩形是否完全覆盖大矩形；
- 如果恰好有一个矩形未被钦定，需要计算剩下这个矩形完全覆盖剩余部分（也是矩形区域）的方案数；
- 如果恰好有两个矩形未被钦定，需要进一步讨论这两个矩形覆盖剩余部分的情况，这里只讨论钦定的小矩形和大矩形高度相同的情况，宽度相同的情况可以类似处理：
  1. 如果剩下两个小矩形的高度均小于大矩形的高度，那么只能一上一下覆盖剩余部分，要求两个小矩形宽度均不小于剩余部分的宽度，并且高度之和不小于大矩形的高度；
  2. 如果剩下两个小矩形中有一个矩形能够完全覆盖剩余部分，那么另一个小矩形可以任意放置；
  3. 否则剩下两个小矩形的高度都要和大矩形高度相同，枚举其中一个小矩形覆盖剩下两个角，再计算剩下这个矩形完全覆盖剩余部分的方案数即可。

A. Intro: Dawn of a New Era

题目要求共享颜色的相邻场景数尽可能多，也就是不共享颜色的相邻场景数尽可能少，进而可以转化为共享颜色的连续场景区间的个数尽可能少。

基于以上转化，我们可以构建这样一个图，其中把每个颜色编号看作一个节点，然后对于每个场景的所有非最大编号的颜色，都向这个场景的最大编号颜色连一条有向边，那么我们要做的实际上就是用尽可能少的路径去覆盖 n 个场景的最大编号颜色所对应的节点的集合，其中还得保证每个场景对应的若干条边至多只有一条在最后的覆盖方案中。

A. Intro: Dawn of a New Era

考虑如何求这样的路径覆盖方案。这就是一个比较经典的网络流问题。可以考虑给每个颜色编号拆一个入点和出点，其中最大编号颜色所对应的节点的入点到出点的流量至少是 1，然后给每个场景再拆一个点，来限制这个场景的非最大编号颜色点连向最大编号颜色点的总流量为 1，最后令源点连向所有入点，所有出点连向汇点，这样问题就变成求一个上下界最小可行流了。由于本题保证流量只会从小编号颜色点流向大编号颜色点，不会形成环流，性质比较优秀，所以可以直接套用相应的模板。

考虑用路径覆盖方案导出最终的排列方案。首先路径中的每条边都对应着一个场景，所以可以先把路径转化成为场景序列，然后对于没有出现在路径中的场景，我们可以根据其最大编号颜色分组，然后找到一个出现在路径中且最大编号颜色相同的场景，然后把同组的没有出现在路径中的场景塞在对应场景后面即可。最后再把所有序列连成一个排列即可。

F. Ursa Minor

考虑 $a_1, a_2, \ldots, a_n$ 和 $b_1, b_2, \ldots, b_m$ 确定的关卡能通关的条件。由裴蜀定理，b 序列实际上能缩成单个 $k = \gcd(n, b_1, b_2, \ldots, b_m)$ 的工具。那么设 $s_t = \sum_{i \bmod k = t} a_i$，我们断言通关的充要条件是 $s_0 = s_1 = \cdots = s_{k-1}$。

必要性显然，一次操作 s_t 相对值不变。对于充分性，逐一操作出 $a_1, a_2, \ldots, a_{n-k}$，因为 s_t 相等，那么合理操作区间 $[n-k+1, n]$ 即能使最后 k 个同时归位。
任取 x, p，在模 p 意义下，定义哈希值 $\sum_i a_i x^{i \bmod k}$，能通关基本等价于该值等于 $s_0(x^0 + x^1 + \cdots + x^{k-1})$，错误概率约 10^{-9}（若双重哈希，错误概率仅 10^{-18}）。

F. Ursa Minor

对于原问题，套用上述通关的判断方法，前半部分得到 k 需要用 ST 表 $O(N \log^2 N)$ 预处理 $O(Q \log N)$ 查询区间 gcd。后半部分由于得到各询问 s_0 的值是经典的值域分块问题，故考虑值域分块。不妨假设 N, Q 同阶：

- 若 $k \le \sqrt{N}$：仅有 $O(\sqrt{N})$ 种不同的 k，对所有 k 持续用 结构I 维护模 p 意义下 $A_i x^{i \bmod k}$ 的区间和。结构I 共有 $O(N\sqrt{N})$ 次修改和 $O(N)$ 次查询，因此使用 $O(1)$ 修改 $O(\sqrt{N})$ 查询的分块结构。
- 若 $k > \sqrt{N}$：仅有 $O(\sqrt{N})$ 种不同的 $i - (i \bmod k)$，而 $x^{i \bmod k} = \frac{x^i}{x^{i-(i \bmod k)}}$，用 结构II 维护模 p 意义下 $A_i x^i$ 的区间和，求解哈希值可以拆分为若干区间和除以 $x^{i-(i \bmod k)}$ 的总和。结构II 共有 $O(N)$ 次修改和 $O(N\sqrt{N})$ 次查询，因此使用 $O(\sqrt{N})$ 修改 $O(1)$ 查询的分块结构。

综上，能够以 $O(N\sqrt{N} + N \log^2 N)$ 的时间复杂度通过此题。

G. Military Maneuver

问题即为在矩形 $[x_l, x_r] \times [y_l, y_r]$ 内随机选一个点 $Q(x, y)$，给定的 n 个点里到 Q 的最远点和最近点分别是 $P_1(x_1, y_1)$ 和 $P_2(x_2, y_2)$，要求 $P_1^2 - P_2^2 = 2x(x_2 - x_1) + 2y(y_2 - y_1) + (x_1^2 - x_2^2 + y_1^2 - y_2^2)$ 的期望。

考虑枚举 P_1 和 P_2，满足条件的 Q 的区域是 $O(n)$ 个半平面求交得到的（凸）多边形区域 D，要对面积为正的 D 计算 $s_x = \int_D x \mathrm{d}\sigma$，$s_y = \int_D y \mathrm{d}\sigma$ 以及 $s = \int_D \mathrm{d}\sigma$，其中 s 即为 D 的面积，s_x 和 s_y 可以通过求出 D 的重心坐标 (b_x, b_y) 之后利用 $b_x = \frac{s_x}{s}$ 以及 $b_y = \frac{s_y}{s}$ 求得。

G. Military Maneuver

直接枚举两个点之后做半平面交的复杂度是 $O(n^3 \log n)$，不能接受。如果对每个 P_i 分别预处理出距离 P_i 相对其他给定点最近的点的区域 V_i 以及距离 P_i 相对其他给定点最远的点构成的区域 F_i，实际上就是最近点和最远点 Voronoi Diagram，所有 V_i 的边数之和为 $O(n)$，所有 F_i 的边数之和也为 $O(n)$，这样在枚举最近点是 P_i、最远点是 P_j 时，只需要将 V_i 和 F_j 求交，复杂度是 $O(n^2 \log n)$。

L. Rook Detection

显然二者必满足其一：棋盘每行有车、棋盘每列有车。不妨先假设前者成立。

若能单独剥离出 $1 \times n$，找到其内的车容易用二分区间解，难点在于去除各行间的影响。

若 $m \times n$ 的棋盘每行有车且第一列确定无车，那么能将第一列的格子作为“检测位”对每行同时二分。故提出以下算法

L. Rook Detection

Step 1：询问 1 次，升高除了左上角的第一列和主对角线。

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

若一行的对角线 1 受控制，我们确定该行的第一列和主对角线必有一车，称其为"二择行"，否则均无车，称其为"二分行"。

L. Rook Detection

Step 2：询问 1 次，升高二择行的第一列、左上角和任意二分行（若无则特判）的对角线和其第一行的投影。

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

二分行的第一列已作为检测位，若其不受控制，我们推翻棋盘每行有车的假设不成立，并且我们凑巧找到了这一行均能作为检测位，旋转下调用之前的二分即可。此外，容易确定第一行的第一列能否作为检测位。

L. Rook Detection

Step 3：询问 log 次，二分行调用之前的二分，合理调整每次二分的方向，能把二择行的对角线塞到 1 层并在不受影响的情况下得知该行第一列、对角线哪个有车。

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & \mathbf{0} \end{bmatrix}$$

（加粗的 **0** 暂时还塞不到 1 层因为会受二分段的影响）