2026 山东省大学生程序设计竞赛

SUA 程序设计竞赛命题组

2026 年 5 月 24 日

概况

- 题目设计：
  - 基础代码能力：I、G、E、H
  - 基础算法应用：K、C、L、D
  - 进阶算法应用：A、J、M、B、F

I. 版本号

题意

- 给定 n 个数对 (x, y)，问先比较第一维，再比较第二维，哪个最大。

- 按题意模拟。
- 复杂度 $\mathcal{O}(n)$。

G. 吸血鬼爬行者

题意

- 有 n 张卡牌，每张卡牌有费用 a_i 和伤害 b_i。把所有牌按一定顺序打出。若这张牌的费用比上一张牌多 1，则伤害倍率加 1，否则伤害倍率变回 1。这张牌造成的伤害是 b_i 乘以倍率。最大化总伤害。
- $n \le 2 \times 10^5$。

- 贪心：把所有牌放到一个网格里，网格的列是费用，同一列的牌（费用相同的牌）按伤害从大到小排序。最后一行行打出来即可。
- 复杂度 $\mathcal{O}(n \log n)$。

G. 吸血鬼爬行者

证明

- 设 c_a 表示费用为 a 的牌有几张，设 $n_{a,k}$ 表示费用为 a 的牌能形成几个长度至少为 k 的连击串，则 $n_{a,k} \le \min(c_{a-k+1}, c_{a-k+2}, \cdots, c_a)$。
- 设 $b_{a,i}$ 表示费用为 a 的牌里，位于第 i 行的牌，则费用为 a 的牌能造成的伤害等于 $\sum\limits_{k=1}^{c_a}\sum\limits_{i=1}^{n}{}_{a,k}b_{a,i}$。
- 在我们的排列方式中，$n_{a,k}$ 都能取到上界，且 $n_{a,k}$ 随着 k 的增大而减小，根据排序不等式，$b_{a,i}$ 也应该随着 i 的增大而减小。

E. 简单的构造题

题意

- 把 0 到 $(nm-1)$ 填入一个 $n \times m$ 的网格，使得对于每个 i，$(i+1) \bmod nm$ 都不和它相邻。
- $3 \le nm \le 2 \times 10^5$

- 可以证明 $nm \le 4$ 无解，也可以通过暴力枚举验证。
- $n = 2$，$m = 3$（或反过来）需要特殊构造。
- 其它情况，把网格看成棋盘格，从 0 开始顺序填。先把黑格子从左上到右下填，再把白格子从左上到右下填。
- 复杂度 $\mathcal{O}(nm)$。

题意

- 给定一张 n 个点 m 条边的无向图，有边权。求一棵边权最小的生成树，使得指定的 k 个点是叶子。
- $n, m \leq 2 \times 10^5$

- 先求剩下的 $(n-k)$ 个点的最小生成树。
- 指定的 k 个点都得连到这棵树上，每个点选边权最小的一条边即可。
- $n = 2$ 特判一下。复杂度 $\mathcal{O}(n + m\log m + m\alpha(n))$。

C. 会议日程

题意

- 给定 n 个区间，将某个区间的一个端点从 x 改成 x' 的代价是 $(x-x')^2$。用最小的代价调整区间，使得区间两两相交，且区间端点仍是整数。
- $n \le 2 \times 10^5$，值域 $0 \le V \le 10^6$。

- 调整后的区间两两相交等价于所有区间有公共点。
- 就是要找到一个整点 x，使所有区间都包含该点。
- 单个区间的代价，是关于 x 的分段凸函数。凸函数相加还是凸函数，所以可以三分 x 找到答案。每次三分后 $\mathcal{O}(n)$ 计算代价即可，不需要维护函数长什么样。
- 复杂度 $\mathcal{O}(n \log V)$。

H. 拼图

题意

- 给定 $n \times m$ 的网格（有的格子是障碍）和 k 块四连通的拼图，拼图不能旋转不能翻转。问选择若干块拼图把网格拼满的方案数。
- $n, m, k \leq 8$

- 深度优先搜索：每次找到左上方第一个空格子（先找有空格子的行，再找最前面的列），枚举把哪个拼图的左上角拼上去即可。
- 复杂度 $\mathcal{O}(k! \times nm)$。

L. 分数迭代

题意

- 给定一个最简分数 a/b（即 $\gcd(a, b) = 1$）。每一秒，分子和分母同时加 1，然后立即化简为最简形式。q 次询问从初始状态开始经过恰好 k 秒后的分数。有 T 组数据。
- $1 \le T \le 10^3$，$1 \le a, b \le 10^{12}$，$1 \le q \le 100$，$1 \le k \le 10^{18}$。

L. 分数迭代

- 如果初始的 $a = b\ (= 1)$，分数永远是 $1/1$，特判即可。
- 否则考虑每次找一个最小的 t 使得 $g = \gcd(a+t, b+t) > 1$，然后分子分母同时约去 g。
- 因为 g 能整除 $(a+t)$ 和 $(b+t)$，那么它也能整除 $|(a+t)-(b+t)| = |a-b| = d$。对 d 分解质因子之后枚举质因子，可以找到最小会发生约分的 t。
- 每次约分之后 d 会变成自己的因子，不需要再重新分解质因子，枚举一开始分解出来的质因子即可。
- 分解质因子需要预处理 $\le 10^6$ 的质数，单组数据的时间复杂度是 $\mathcal{O}(\frac{\sqrt{n}}{\log n} + q\log^2 n)$。

题意

- 令 $f(x)$ 为正整数 x 在十进制表示下的最大数码，给定 l_a，r_a，l_b 和 r_b，计算 $\sum_{a=l_a}^{r_a}\sum_{b=l_b}^{r_b} f(a+b)$。有 T 组数据。
- $1 \le T \le 10^3$，$1 \le l_a \le r_a \le 10^9$，$1 \le l_b \le r_b \le 10^9$。

- 令 $g(n, m) = \sum_{a=1}^{n}\sum_{b=1}^{m} f(a+b)$，那么有

$$
\begin{aligned}
\sum_{a=l_a}^{r_a}\sum_{b=l_b}^{r_b} f(a+b) = g(r_a, r_b) - g(l_a-1, r_b) \\
- g(r_a, l_b-1) + g(l_a-1, l_b-1)
\end{aligned}
$$

D. 最大数码 2

- 使用数位 DP 求解 $g(n, m)$，考虑 $max(len(n), len(m)) + 1$ 位数（高位补零）。
- $f[i][j][0/1][0/1][0/1]$ 表示同时考虑 a 和 b 的最高 i 位数，$a + b$ 在这 i 位的最大数码是 j，a 这 i 位是否等于 n 这 i 位，b 这 i 位是否等于 m 这 i 位，$a + b$ 的第 i 位是否要向下借位，的方案数。
- 转移的时候枚举 a 和 b 在第 $i + 1$ 位的值，以及 $a + b$ 的第 $i + 1$ 位是否要向下借位。
- 最后将合法状态（最低位不能再向下借位）的方案数乘上状态对应的 j 再求和即可。

A. 华容道

题意

- 给定 $n \times m$ 的网格，每个格子里有一个箭头。只有当箭头所指方向没有其它箭头时，才能移除它。对于每个箭头，求为了移除它，至少要移除几个箭头。
- $n \times m \le 5 \times 10^4$。

- 考虑每个箭头向该方向上所有其它箭头连边，问题就转化为有向图上求每个点的后继数量。但是边数在最坏情况下可以达到 $\mathcal{O}(nm(n+m))$。
- 其实每个箭头连边到和它一样的即可停止，边数是 $\mathcal{O}(nm)$。
- 例子：$\uparrow\leftarrow\downarrow\leftarrow$，因为箭头 2 已经连了 2 之前的所有箭头，所以箭头 4 只要从箭头 2 开始连边即可，不需要再往前连。
- 利用 bitset 求有向图求后继数量的复杂度为 $\mathcal{O}(NM/w)$，其中 N 是点数，M 是边数。因此本题复杂度是 $\mathcal{O}((nm)^2/w)$。

J. 插松枝

题意

- 有两个栈，其中一个栈里有 n 个元素，另一个栈一开始是空的。
- 每次操作可以选一个栈删除栈顶，或将一个栈顶元素移到另一个栈顶。
- 必须按从大到小的顺序删除元素，求最少操作次数。
- $n \leq 4 \times 10^5$

- 问题可以转化为：给一个序列和一条分界线，一开始分界线在第一个元素的左边，每次操作可以删除分界线左/右的一个元素，或让分界线往左/右移动一位。

J. 插松枝

- 设当前最大元素为 x，必须要把所有 x 都删除，然后才能删除其它元素。
- 关键观察：最优方案删完所有 x 时，分界线一定位于原来第一个 x 或最后一个 x 的位置。因为最优删除方案肯定是从中间先删到一端再删到另一端，不可能故意留着中间不删，回来再删。
- 因此维护 $f(i, 0/1)$ 表示删完 i 后，分界线位于原来第一个/最后一个 i 时的最少操作次数，从 $f(i+1, 0/1)$ 转移过来。转移时，需要求一个区间内还剩几个元素，用树状数组维护即可。
- 复杂度 $\mathcal{O}(n \log n)$。

M. 博物馆奇妙夜 2

题意

- 给定 n 个点构成的闭合折线表示巡逻路线，保安沿 $1 \to 2 \to \cdots \to n \to 1 \to \cdots$ 的顺序巡逻。当保安不在线段端点时，其视野是无限延伸的圆心角为 $2a$ 的圆锥，且视野中线方向与当前巡逻线段的前进方向相同。给定 m 件展品的位置，求能够看到所有展品的巡逻路线的总长度。
- $3 \le n \le 2 \times 10^5$，$1 \le m \le 2 \times 10^5$，$0 < a < 90$。

- 设保安 P 所在线段 AB 的单位方向向量为 $\vec{u}$，巡逻视野左右边界的单位方向向量分别为 $\vec{v_L}$ 和 $\vec{v_R}$（$\vec{u}$ 旋转 $\pm a$ 得到）。
- 由于 $0 < a < 90$，展品 Q 从 P 可见当且仅当向量 $\overrightarrow{PQ}$ 在 $\vec{v_L}$ 右侧且在 $\vec{v_R}$ 左侧。也就是

$$\vec{v_L} \times (Q - P) \le 0 \quad \text{且} \quad \vec{v_R} \times (Q - P) \ge 0$$

- 设 $P = A + t\vec{u}$ $(0 < t < |AB|)$，则有

$$t \le \frac{\vec{v_L} \times (Q - A)}{\vec{v_L} \times \vec{u}} \quad \text{且} \quad t \le \frac{\vec{v_R} \times (Q - A)}{\vec{v_R} \times \vec{u}}$$

- 注意这里 $\vec{v_L} \times \vec{u} < 0$，$\vec{v_R} \times \vec{u} > 0$。

- 考虑所有展品的限制，就是要求

$$\min\left(\frac{\max_{i=1}^{m}(\vec{v_L}\times Q_i)-\vec{v_L}\times A}{\vec{v_L}\times\vec{u}},\ \frac{\min_{i=1}^{m}(\vec{v_R}\times Q_i)-\vec{v_R}\times A}{\vec{v_R}\times\vec{u}}\right)$$

- 求解 $\max_{i=1}^{m}(\vec{v_L}\times Q_i)$ 和 $\min_{i=1}^{m}(\vec{v_R}\times Q_i)$ 是经典问题。对所有 Q_i 分别求出上凸壳和下凸壳，在两个凸壳上分别三分出最值即可，时间复杂度是 $\mathcal{O}((n+m)\log m)$。

B. 字典 2

题意

- 给一棵以 1 为根的树，每个节点上有一个小写英文字符 c。对于节点 u，从 u 到根路径上的字符依次拼接得到字符串 $S(u)$，令 $\mathcal{D}$ 为所有 $S(u)$ 的非空子串的（不可重）集合，需要支持**在线**加叶子，并在 $S(\text{新加叶子})$ 的非空子串加入 $\mathcal{D}$ 之后，回答 $\mathcal{D}$ 中有多少字符串的字典序不大于 $S(\text{新加叶子})$。
- $1 \le n, q \le 10^5$。

- 如果将所有 $S(u)$ 按字典序排序，记字符串 $S(u)$ 与 $S(pre_u)$ 的最长公共前缀长度为 $LCP(u, pre_u)$，那么 $\mathcal{D}$ 中字典序不大于 $S(v)$ 的字符串数量是

$$\sum_{S(u) \le_{lex} S(v)} (|S(u)| - LCP(u, pre_u))$$

B. 字典 2

- 使用平衡树维护所有 $S(u)$ 的顺序，现在新增一个叶子 v，要实现 $S(v)$ 与其他 $S(u)$ 的比较：
  - 如果 $c_v \neq c_u$，直接比较 c_v 和 c_u；
  - 否则考虑 $S(v) = c_v + S(fa_v)$，$S(u) = c_u + S(fa_u)$，只需要比较 $S(fa_v)$ 和 $S(fa_u)$，这两个字符串之前已经维护在平衡树里，可以在平衡树上找到排名。
- 如果能在平衡树上每个节点维护一个实数区间 (l, r)，左儿子对应 $(l, \frac{l+r}{2})$，右儿子对应 $(\frac{l+r}{2}, r)$，要比较平衡树上两个节点的先后顺序，只需要取出区间中点比较数值。
- 使用替罪羊树维护即可，子树不平衡时重构子树，同时重算整个子树的实数区间。由于树高是 $\mathcal{O}(\log(n+q))$，浮点精度不需要特别高。

B. 字典 2

- 为了支持求和，还要在平衡树上维护 $LCP(u, pre_u)$。
- 插入 v 之前，查出前驱 pre_v 和后继 suc_v，要求 $LCP(v, pre_v)$ 以及 $LCP(v, suc_v)$。
- 与比较字典序的方式类似，要求 $LCP(v, u)$：
  - 如果 $c_v \neq c_u$，那么 $LCP(v, u) = 0$；
  - 否则 $LCP(v, u) = 1 + LCP(fa_v, fa_u)$，如果 $fa_v = fa_u$ 就是求字符串长度，否则在平衡树上找到 fa_v 和 fa_u 的排名 rk_v 和 fa_u，在排名区间 $(\min(rk_v, rk_u), \max(rk_v, rk_u)]$ 查询 LCP 的最小值就是 $LCP(fa_v, fa_u)$。
- 时间复杂度是 $\mathcal{O}((n + q) \log(n + q))$。

F. 礼物归位

题意

- 给定两个不同的排列 p 和 q，每次操作可以交换一组 p_i 和 $p_{i \bmod n+1}$，要用最少的操作使得 $p = q$，输出一组方案。
- $3 \le n \le 3\,000$。

算法 数学 排序 排列组合

关注者 10 被浏览 852

给定一个无序数列，假定一次操作可使相邻或首尾两数交换，至少需要多少次操作使之有序?

给定一个n长数列（数列中无相同元素），
假定一次操作可以使相邻两数交换或者使首尾两数交换。
至少需要多少次操作可以使得数列从小到大有序排列。

若只允许相邻两数交换，那么操作次数就是逆序数，可是如果也允许首尾交换呢？

关注问题 写回答 邀请回答 好问题 3 添加评论 分享 修改问题 ··· 收起

F. 礼物归位

- 一个符合直觉的猜想：存在一个断点，交换不经过这个断点。如果猜想成立，可以枚举 $\mathcal{O}(n)$ 断点，然后 $\mathcal{O}(n\log n)$ 计算逆序对，找出最优的断点转换成序列上的问题。
- 但是这个猜想并不成立：

[redacted] ...

n=10时
7 9 1 5 3 0 2 4 6 8
按这个方法需要14次置换完成
7 9 5 1 3 0 2 4 6 8
则需要19次置换完成

而这两个序列显然可以通过一次置换转化
2024-12-18 · 上海 回复 喜欢

F. 礼物归位

- 设 x_i 为从当前状态操作到目标状态时位置 i 的元素所需的位移量，那么总有 $\sum_{i=1}^{n} x_i = 0$。
- 记 $pos_q[v]$ 为值 v 在 q 中的位置，那么初始时

$$x_i \equiv pos_q[p_i] - i \pmod n$$

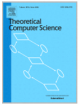
- 初始化 $x_i = pos_q[p_i] - i$，如果存在 i 和 j 使得 $x_i - x_j > n$，令 $(x_i, x_j) \leftarrow (x_i - n, x_j + n)$ 可减少一次元素移动过程中的交叉。
- 最后得到一组 x_i 对应了最少操作次数时每个元素的位移量。每个元素最多被改变一次，朴素实现的时间复杂度是 $\mathcal{O}(n^2)$。

F. 礼物归位

ELSEVIER

Theoretical Computer Science

Volume 36, 1985, Pages 265-289



The complexity of finding minimum-length generator sequences

Mark R. Jerrum

Show more

Add to Mendeley Share Cite

https://doi.org/10.1016/0304-3975(85)90047-7

Get rights and content

F. 礼物归位

- 考虑构造方案，如果 $p \neq q$ 那么存在 $x_i > x_{i \bmod n+1}$，也就是当前 p_i 的目标位置比 $p_{i \bmod n+1}$ 的更“靠右”，这两个元素在移动过程中必有一次交叉，现在交换 p_i 和 $p_{i \bmod n+1}$ 就实现了这次交叉，然后令 $(x_i, x_{i \bmod n+1}) \leftarrow (x_{i \bmod n} + 1, x_i - 1)$。
- 队列维护交换的候选，取出队首 i，如果 $x_i > x_{i \bmod n+1}$ 就交换，并将 i 的相邻两个位置入队。由于操作次数是 $\mathcal{O}(n^2)$，时间复杂度是 $\mathcal{O}(n^2)$。

Thank you!