# 2023 广东省大学生程序设计竞赛

SUA 程序设计竞赛命题组

2023 年 5 月 14 日

# 概况

- 阶段一（基本编程技巧）：A（枚举）、C（循环）、K（搜索）、I（二分）。
- 
- 阶段二（常见算法应用）：D（贪心）、M（几何）、B（单调队列）、E（字典树）、F（线段树）。
- 
- 阶段三（思维能力测试）：J（数学）、H（建图）、G（思维）、L（图论）。

# A. 算法竞赛

## 题意

- 给一个序列 $s_1, s_2, \cdots, s_n$，给两个整数 $y_1$ 和 $y_2$，求两个整数之间有多少整数不在序列里。

- 枚举每个整数并检查即可，复杂度 $\mathcal{O}(n + y_2 - y_1)$。

**题意**

- 有 $n$ 间商店，第 $i$ 间商店的买卖价格都是 $a_i$，且最多交易 $b_i$ 次，问最大获利。

- 每次应该从价格最便宜的商店购买货物，并卖给价格最贵的商店。用双指针模拟这一贪心策略即可。
- 复杂度 $\mathcal{O}(n \log n)$，主要是排序的复杂度。

# K. 独立钻石

## 题意

- 给定独立钻石棋的规则：选择一枚棋子，将它跳过相邻棋子到空格上，并移除被跳过的棋子。
- 给定初始局面，求最后最少能剩几枚棋子。

- 每个棋子只能被横向或者纵向跳过，因此当棋盘上存在 $k$ 枚棋子时，共有 $2k$ 种操作可以选择。
- 每一步都将减少 $1$ 枚棋子，因此至多执行 $(k-1)$ 步。
- 直接通过 dfs 进行搜索的复杂度为 $\mathcal{O}(T \times nm \times \prod_{i=2}^{k} 2i) \approx 1.7 \times 10^7$，无需任何优化即可在时限内通过。

# I. 路径规划

## 题意

- $n \times m$ 网格图上，$0$ 到 $(nm-1)$ 的所有整数恰好出现一次。
- 求一条从左上角到右下角的路径，每次只能往右或往下，使得路径的 mex 最大。

- 每一步只能往右或者往下走，因此将路径上每个格子的坐标按行为第一关键字，列为第二关键字排序后，排在前面的坐标的列编号，一定小于等于排在后面的坐标的列编号。
- 二分答案 $x$，将从 $0$ 到 $(x-1)$ 的每个整数所在的格子的坐标排序，并检查列编号是否满足以上条件。
- 实际上不需要排序函数，按顺序枚举并检查即可，复杂度 $\mathcal{O}(nm\log(nm))$。

# D. 新居规划

## 题意

- 有 $n$ 个人和一排 $m$ 间房，每个人住进一间房子里。第 $i$ 个人如果有邻居满意度为 $a_i$，无邻居满意度为 $b_i$。求最大总满意度。
- 如果两个人住进房子 $x$ 和 $(x+1)$，他们互为邻居。

# D. 新居规划

- 如果已知 $k$ $(2 \le k \le n)$ 个人有邻居，剩下的人没有邻居，怎样选择有邻居的人才能使总满意度最大化？
- 这是一个经典问题。先假设所有人都是没邻居的，得到总满意度 $\sum_{i=1}^{n} b_i$。当第 $i$ 个人从没邻居变成有邻居时，总满意度将增加 $(a_i - b_i)$。因此选择 $(a_i - b_i)$ 最大的 $k$ 个人变成有邻居的即可。排序后可以在 $\mathcal{O}(n)$ 的复杂度内一次性算出 $k = 2, \cdots, n$ 的最大总满意度。

# D. 新居规划

- 如果 $k$ 个人有邻居，剩下的人没有邻居，这样的布局至少需要 $k + 2(n - k) = 2n - k$ 栋房子（即有邻居的人都住在最左边，然后每隔一栋房子住一个没邻居的人）。因此只有满足 $2n - k \leq m$ 才能考虑。
- 最后，别忘了考虑所有人都没有邻居的情况。这要求 $m \geq 2n - 1$。
- 复杂度 $\mathcal{O}(n \log n)$。主要是排序的复杂度。

# M. 计算几何

**题意**

- 将一个凸多边形沿着两个顶点的连线分成两块，最小化两个小多边形直径平方的和。

- 枚举用于切开大多边形的顶点 $i$ 和 $j$，问题变为如何快速计算两个小多边形的直径。显然两个小多边形也都是凸多边形。

# M. 计算几何

- 凸多边形的直径一定是某两个顶点的连线，维护 $f(i,j)$ 表示第 $i$ 个顶点到第 $j$ 个顶点之间，两个顶点之间的最大距离的平方（如果 $i>j$ 那就是顶点 $i,i+1,\cdots,n,1,2,\cdots,j$ 之间的最大距离）。
- 令 $\mathsf{dis}(i,j)$ 表示顶点 $i$ 和 $j$ 之间的距离，容易得到区间 dp 方程

$$f(i,j)=\max\{f(i+1,j),f(i,j-1),\mathsf{dis}^2(i,j)\}$$

  初值 $f(i,i+1)=\mathsf{dis}^2(i,i+1)$。复杂度 $\mathcal{O}(n^2)$。
- 两个小多边形的直径平方和即为 $f(i,j)+f(j,i)$，取最小值为答案即可。
- 当然，要求顶点 $i$ 和 $j$ 的连线切到凸多边形内部。根据凸多边形的性质，这等价于顶点 $j$ 不能在顶点 $i$ 和 $(i+1)$ 的连线上，也不能在顶点 $i$ 和 $(i-1)$ 的连线上。算出叉积进行判断即可。

# B. 基站建设

## 题意

- $n$ 个点排成一行，选择每个点都有一个代价。给 $m$ 个区间，要求每个区间里至少选一个点，求最小总代价。

- 维护 $f(i)$ 表示只考虑前 $i$ 个点，且第 $i$ 个点必选的最小总代价。考虑上一个点选的是 $j$，得到 dp 方程：
- $f(i) = \min_{j} f(j) + a_i$
- 因为每个区间里至少选一个点，因此 $[j+1, i-1]$ 之间不能存在一个完整区间。
- 对于每个 $1 \le i \le n$，计算 $p_i$ 满足 $[p_i, i]$ 之间不存在一个完整区间，且 $p_i$ 尽可能小，则 $j \ge p_{i-1} - 1$。
- 所有 $p_i$ 可以用双指针算出来，dp 方程可以用单调队列优化。复杂度 $\mathcal{O}(n+m)$。

# E. 新怀质问

**题意**

- 给 $n$ 个字符串，从中选出 $k$ 个，使得其中两个字符串 lcp 的最大值最小（按字典序比大小）。

# E. 新怀质问

- 假设我们已经确定了答案的前三位是 abc，接下来要确定第四位。为了让字典序尽量小，我们需要从 a 到 z 枚举第四位。
- 假设我们枚举到了 d，我们考虑已知答案为 abcd* 时，与已知答案为 abc* 时相比，能选择的字符串数量如何变化。
  - 所有 abc[a-d]* 都能选择，因为它们的最长公共前缀肯定小于等于 abcd*。
  - 所有 abc[e-z]* 的字符串，原来答案是 abc* 的时候都能选择，现在每种字母只能选一个，否则比如 abce* 选了两个，那答案就至少是 abce > abcd* 了。
  - 剩下的字符串可选情况维持不变。
- 如果我们能选至少 $k$ 个字符串，那么答案的第四位就是 d，否则我们要继续枚举 e、f、...

# E. 新怀质问

- 确定了答案的第四位以后，我们还要确定答案是不是只有四位。考虑已知答案为 abcd 时，与已知答案为 abcd* 时相比，能选的字符串数量如何变化。
  - 所有 abcd[a-z]* 的字符串，原来答案是 abcd* 时都能选择，现在每种字母只能选一个，否则答案大于 abcd。
  - 剩下的字符串可选情况维持不变。
- 如果我们能选至少 $k$ 个字符串，那么答案就只有四位，否则继续枚举第五位。
- 在 trie 上计算这一过程即可。复杂度 $\mathcal{O}(26 \times \sum |s|)$，

# F. 格子旅行

## 题意

- 给一行 $n$ 个格子，每个格子有个颜色，还有个权值。
- 每次操作可能修改颜色和权值。
- 每次询问给定一个颜色集合 $\mathbb{A}$ 和起点格 $s$，要求从 $s$ 出发，只能走 $\mathbb{A}$ 里的颜色，求能走到的格子的权值和。

# F. 格子旅行

- 显然旅行的范围是包含起点的连续区间。每次询问，我们通过二分找出旅行的左右端点，然后询问该区间的权值之和即可。
- 为了通过二分找出旅行的端点，我们需要快速求出一个区间里所有的颜色是否都在 $\mathbb{A}$ 里。也就是说，求 $\mathbb{A}$ 中所有颜色在区间中出现次数之和，是否等于区间长度。
- 维护一个线段树 / 树状数组，每个节点保存一个哈希表，表示该区间中出现了哪些颜色，以及每种颜色出现了几次即可。
- 如果先二分答案，然后再算颜色出现次数之和的复杂度是 $\mathcal{O}(\sum k\log^2 n)$，需要较小的常数才能通过本题。正确的做法应该是直接在线段树 / 树状数组上进行二分 / 倍增，复杂度 $\mathcal{O}(\sum k\log n)$。

# J. X 等于 Y

**题意**

- 给正整数 $x$，$y$，$A$，$B$，求 $a$ 和 $b$ 满足 $2 \leq a \leq A$，$2 \leq b \leq B$，且 $x$ 在 $a$ 进制下的表示与 $y$ 在 $b$ 进制下的表示相同。

- 当序列长度为 1 时，要求 $x = y$，此时 $a = b = 2$ 即可。
- 当序列长度大于等于 3 时，$a \leq \sqrt{x}$，$b \leq \sqrt{y}$，计算每种 $a$ 和每种 $b$ 的答案，看是否有一样的即可。可以用哈希表记录，但是常数比较大，也可以用双指针的方式判断。这种情况的复杂度为 $\mathcal{O}(\sqrt{x} \times \log x)$。

# J. X 等于 Y

- 当序列长度为 $2$ 时，设最高位为 $t$，有 $t \le \sqrt{x}$ 且 $t \le \sqrt{y}$（否则如果 $t > \sqrt{x}$，因为 $t < a$，$x \ge ta > x$ 矛盾）。有以下式子：
  - $1 \le t < a$，$1 \le t < b$：高位不能超过进制基数。
  - $0 \le x - ta < a$，$0 \le y - tb < b$：低位不能超过进制基数。
  - $x - ta = y - tb$：低位也要一样。
  - $2 \le a \le A$，$2 \le b \le B$：题目要求。
- 利用第三条等式，把所有 $b$ 代换成 $b = \frac{y-x}{t} + a$，就能得到 $a$ 的范围。因为 $b$ 也要是整数，所以 $(y - x) \bmod t = 0$ 的 $t$ 才能检测。这种情况的复杂度为 $\mathcal{O}(\sqrt{x})$。

# H. 流画溢彩

## 题意

- 有长度为 $n$ 的序列 $a_1, a_2, \cdots, a_n$，一开始序列里都是 $0$。有 $m$ 个操作，第 $i$ 个操作把 $a_{l_i}$ 改成 $x_i$，以及把 $a_{r_i}$ 改成 $y_i$，每个操作都要执行恰好一次。求一个操作顺序使得序列元素总和最大。
- $1 \le x_i, y_i \le 2$。

- 因为后面的操作会覆盖前面的操作，思考起来比较麻烦。我们不妨把操作顺序倒过来，这样一旦某一列的值确定了，后续操作就再也不会更改这一列的值。
- 以下题解中操作的“先后”，是按操作顺序逆转之后的“先后”来说的。
- 首先，如果某个操作把两个数都改成 $2$，那么这个操作肯定最优，最先考虑；相应地，如果某个操作把两个数都改成 $1$，那么这个操作肯定最差，最后考虑。剩下的就是一个数改成 $1$，一个数改成 $2$ 的操作。

# H. 流画溢彩

- 本题的关键在于把题目转化为图论问题。把每一列看成一个点，把每个操作从改成 $1$ 的那一列向改成 $2$ 的那一列连一条有向边。
- 考虑选择一条边 $u \to v$，进行它代表的操作。此时 $a_u$ 的值将被锁定为 $1$，而 $u$ 能直接或间接到达的所有点的值将被锁定为 $2$（只要从 $u$ 出发，按 dfs 序走一遍 $u$ 能到达的边即可）。
- 我们要做的就是让锁定为 $1$ 的列尽量少，也就是选择尽量少的点，让它们能到达的点的并集等于整张图的点集。

# H. 流画溢彩

- 容易发现，将强连通分量缩点以后，我们将得到一个有向无环图。有向无环图上每一个入度为 $0$ 的点我们都必须选择，才能让它们本身以及它们的后续覆盖整张图。也就是说，我们每次从一个入度为 $0$ 的强连通分量中选择一个点，按 dfs 顺序输出边的编号即可。
- 这里有一个细节：如果一个入度为 $0$ 的强连通分量被一个 $(l_i, 2, r_i, 2)$ 操作所影响，那么应该选择被影响的点为 dfs 的起始点，因为被影响的点代表的列已经被锁定为 $2$。

# G. 交换操作

## 题意

- 给定长度为 $n$ 的非负整数序列 $A = a_1, a_2, \ldots, a_n$，定义

$$F(A) = \max_{1 \le k < n} \left( (a_1 \,\&\, a_2 \,\&\, \cdots \,\&\, a_k) + (a_{k+1} \,\&\, a_{k+2} \,\&\, \cdots \,\&\, a_n) \right)$$

  其中 $\&$ 表示按位与操作。
- 可以进行至多一次交换操作：选择两个下标 $i$ 和 $j$ 满足 $1 \le i < j \le n$，交换 $a_i$ 与 $a_j$ 的值。
- 求经过至多一次交换后，$F(A)$ 的最大值。

# G. 交换操作

- 假设已经确定了分界点 $k$，考虑交换哪两个数才有意义。
- 令 $f(i,j) = a_i \,\&\, a_{i+1} \,\&\, \cdots \,\&\, a_j$，称满足 $f(1,i) \neq f(1,i-1)$ 的下标 $i$ 为“前缀关键点”。可以发现，关键点只有 $\log a_i$ 个。同理，称满足 $f(i,n) \neq f(i+1,n)$ 的下标 $i$ 为“后缀关键点”，关键点也只有 $\log a_i$ 个。
- 因此，交换可以被分为三类情况。

# G. 交换操作

- 【交换两个非关键点】如果从一个前缀里拿走一个非关键点，这个前缀的 $\&$ 值不会改变；同理，如果从一个后缀里拿走一个非关键点，这个后缀的 $\&$ 值也不会改变。交换后，由于多 $\&$ 一个数不会让值变大，因此这样的交换没有意义。
- 【交换两个关键点】前后缀关键点分别只有 $\log a_i$ 种，直接枚举即可。这一类的总复杂度为 $\mathcal{O}(n\log^2 a_i)$。

# G. 交换操作

- 【交换一个关键点和一个非关键点】不妨假设交换的是前缀关键点 $i$ 和后缀非关键点 $j$。容易算出交换之后，后缀的 $\&$ 值为 $f(k+1,n)\,\&\,a_i$。也就是说，只要选定了 $i$，无论选哪个 $j$ 都不影响后缀的 $\&$ 值，那么我们选择让交换之后，前缀的 $\&$ 值最大的 $j$ 即可。即最大化 $f(1,i-1)\,\&\,f(i+1,k)\,\&\,a_j$。
- 注意到对于一个固定的关键点 $i$，$f(i+1,k)$ 的值只有 $\log a_i$ 种，而关键点 $i$ 也只有 $\log a_i$ 种，那么 $v=f(1,i-1)\,\&\,f(i+1,k)$ 的值只有 $\log^2 a_i$ 种。因此对于每种 $v$，$\mathcal{O}(n)$ 计算 $g(v,k)$ 表示 $k+1\le j\le n$ 里，$v\,\&\,a_j$ 的最大值即可。这一类的总复杂度也为 $\mathcal{O}(n\log^2 a_i)$。
- 因此本题复杂度 $\mathcal{O}(n\log^2 a_i)$，涉及到对 $f$ 值的计算可以通过 RMQ 在 $\mathcal{O}(n\log n)$ 的复杂度内预处理，后续每次查询只要 $\mathcal{O}(1)$ 的复杂度。

# L. 经典问题

## 题意

- 给一个完全图，其中 $m$ 条特殊边的边权是指定的，其它边 $i-j$ 的边权是 $|i-j|$。求最小生成树。

- 求完全图的最小生成树一般使用 Boruvka 算法。
- 该算法是最早发现的最小生成树算法（1926），因此本题取名《经典问题》。

- 称特殊边连接的节点为特殊点，其它节点为一般点。可以发现，$2m$ 个特殊点将一般点分成了 $(2m+1)$ 段。
- 一般点和其它点的连边权值至少为 $1$。因此根据最小生成树的性质，一段一般点 $l,(l+1),\cdots,r$ 内部是从小到大依次相连的。
- 连接后，我们可以把图简化为 $(2m+1)$ 个 “连续点”（每个连续点代表连续的一段一般点）以及 $2m$ 个特殊点的完全图。
- 根据 Boruvka 算法，问题变为：快速维护每个点向其它连通块连边的最小边权。

# L. 经典问题

- 每次计算 $f_i$ 表示点 $i$ 左边最近的，且和它不在同一连通块的点是哪个。
- 同理，计算 $g$ 表示点 $i$ 右边最近的，且和它不在同一连通块的点是哪个。
- 从左到右枚举每个点。如果该点是连续点，选择 $f_i$ 和 $g_i$ 中距离最近的即可。
- 如果该点是特殊点，则需要向左 / 右枚举到第一个和它没有连边，且不在同一连通块内的点。另外还要考虑与它相邻的所有特殊边。这一步总体是 $\mathcal{O}(m)$ 的。
- Boruvka 算法执行 $\mathcal{O}(\log \text{点数})$ 轮，因此总体复杂度为 $\mathcal{O}(m \log m)$。

# 最后

- 没听明白？没关系。
- 访问 https://sua.ac/wiki/ ，有文字版题解与带注释的参考代码。

# Thank you!