试题讲解

华中农业大学出题组

华中农业大学第十五届程序设计竞赛

2026 年 4 月 4 日

概述

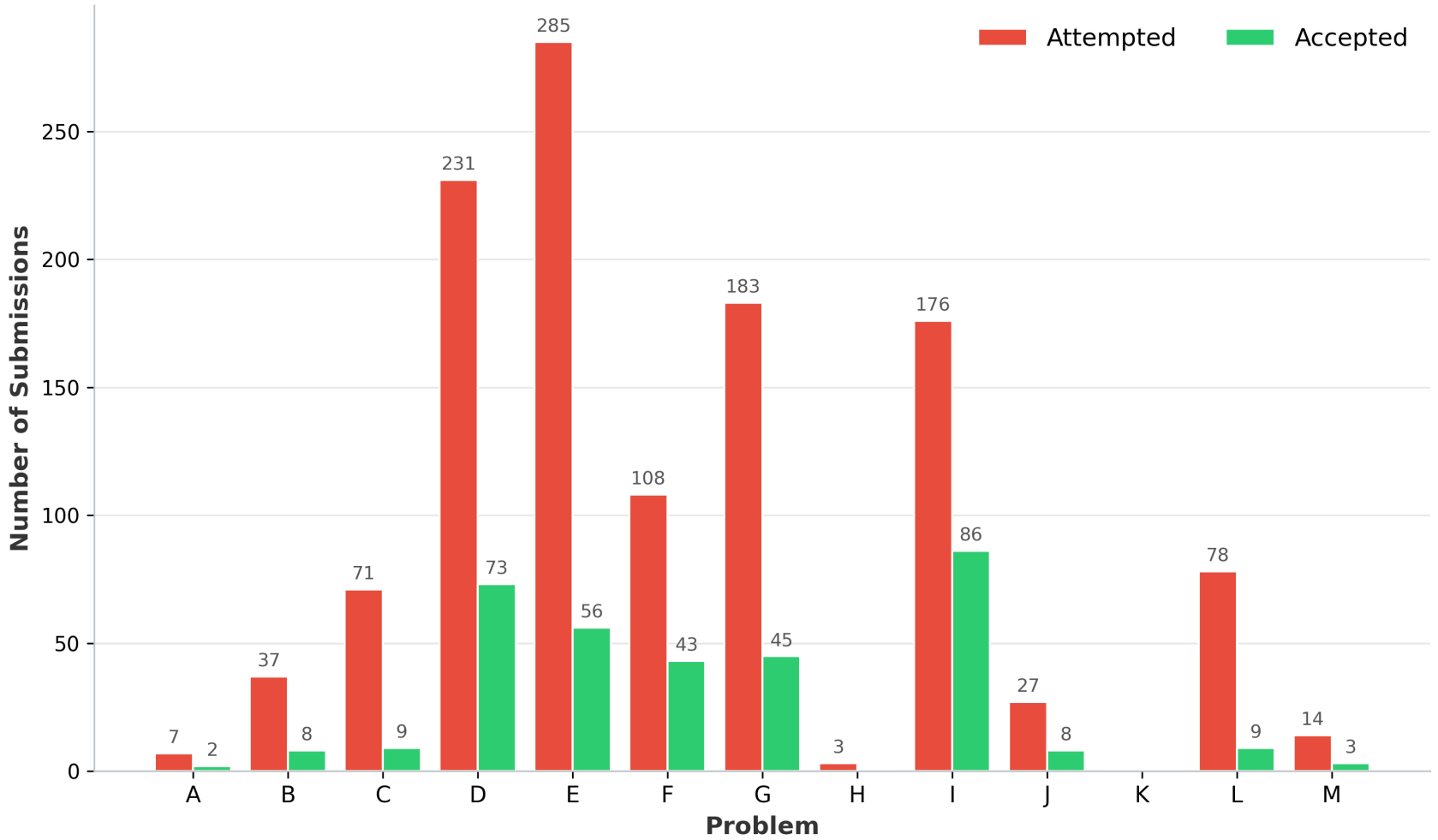
难度预估

- Stage 1: I D F
- Stage 2: G B C
- Stage 3: L J
- Stage 4: A M H K
- Stage 0: E

- 预估难度排序:
  $I < D < F < G < B < E = C < L < J < A < M < H < K$

概述

封榜前，各题提交情况如下：



I. 奶茶

题目大意

需要购买 n 杯奶茶。单杯奶茶 a 元，双人套餐（两杯）b 元。求买恰好 n 杯奶茶的最小总花费。

- 签到题，直接比较 $2a$ 与 b 的大小进行分类讨论：

$$\text{Ans} = \begin{cases} n \times a & (2a \le b) \\ \lfloor \frac{n}{2} \rfloor \times b + (n \bmod 2) \times a & (2a > b) \end{cases}$$

- 时间复杂度 $\mathcal{O}(1)$。

D. 塔菲的 LCM

题目大意

给定正整数 c，求满足 $a+b=c$ 的正整数 a,b，使得 $\operatorname{lcm}(a,b)$ 最大。

- $\max\limits_{a+b=c} \operatorname{lcm}(a,b) = \max\limits_{a+b=c} \dfrac{ab}{\gcd(a,b)}$
- $ab = \left(\dfrac{a+b}{2}\right)^2 - \left(\dfrac{a-b}{2}\right)^2 = \dfrac{c^2}{4} - \dfrac{(a-b)^2}{4}$
- 显然当 c 固定时，差值 $|a-b|$ 越小则乘积 ab 越大。原问题等价于贪心求解：

$$\begin{cases} \min |a-b| \\ \gcd(a,b) = 1 \end{cases}$$

D. 塔菲的 LCM

- 设 $k = \lfloor c/2 \rfloor$。

Case 1: $c \equiv 1 \pmod 2$

$$a = k,\ b = k+1 \implies \gcd(a,b) = 1 \implies \text{Ans} = k(k+1)$$

D. 塔菲的 LCM

Case 2: $c \equiv 0 \pmod 4$ (此时 k 必为偶数)

$$a = k-1,\ b = k+1 \implies \gcd(a,b) = \gcd(2, k-1) = 1$$
$$\implies \text{Ans} = k^2 - 1$$

Case 3: $c \equiv 2 \pmod 4$ (此时 k 必为奇数)

$$a = k-2,\ b = k+2 \implies \gcd(a,b) = \gcd(4, k-2) = 1$$
$$\implies \text{Ans} = k^2 - 4$$

注：需特判边界 $c = 2$ 时，输出 1。时间复杂度 $\mathcal{O}(1)$。

F. 钒钒的括号序列

题目大意

给定一棵由 n 对合法括号构成的二叉树。括号对按左括号出现顺序编号为 $1 \sim n$。对于编号为 p 的括号对（所在位置为 i, j）：

- 若 s_{i+1} 为（，则该左括号所属编号 l 作为 p 的左孩子；
- 若 s_{j+1} 为（，则该左括号所属编号 r 作为 p 的右孩子。

现给出这棵二叉树的结构（根节点保证为 1），求还原出原始序列。

F. 钒钒的括号序列

原题的建树规则，实际上恰好对应了对二叉树的一种特定递归遍历。

DFS 流程

从根节点（编号 1）开始，执行以下操作：

- 进入节点：输出左括号（
- 左子树：若有左孩子，递归访问其左子树
- 节点本身：输出右括号 ）
- 右子树：若有右孩子，递归访问其右子树

时间复杂度 $\mathcal{O}(n)$。

G. 塔菲大战哥布林

题目大意

给定 N 对属性 (A_i, B_i, C_i)，若 $A_i \geq B_i$ 则产生 C_i 收益。Q 次独立询问，每次可将至多一个 A_k 修改为 X，求最大总收益。

- 基础收益为初始状态下满足 $A_i \geq B_i$ 的集合，其总收益记为常数 S。
- 考虑贪心策略，修改已命中的目标无正收益，故修改名额必交由未命中集合 $U = \{i \mid A_i < B_i\}$。
- 问题等价于：针对每次询问 X，在集合 U 中寻找合法的最大增益：

$$\text{Ans} = S + \max_{j \in U,\ B_j \leq X} \{C_j\}$$

G. 塔菲大战哥布林

- 预处理：将集合 U 中的元素按 B_i 升序排序，记为序列 arr。维护 C_i 的前缀最大值 pre 数组：

$$pre[i] = \max(pre[i-1], arr[i].C)$$

- 在线查询：对给定 X，使用 `upper_bound` 找到序列中最后一个满足 $B \le X$ 的位置 pos。
- 若存在合法 pos，对应答案即为 $S + pre[pos]$；否则仍为 S。

时间复杂度：预处理 $\mathcal{O}(N \log N)$，单次查询 $\mathcal{O}(\log N)$。总复杂度 $\mathcal{O}((N+Q)\log N)$。

B. Bus 博弈

题目大意

n 个座位（含坏座），Alice/Bob 轮流落子，要求与已有棋子距离 $> k$。Alice 极大化总人数，Bob 极小化。求最优策略下的最终状态。

- 注意到 $n \leq 20$，状态空间仅 2^{20}，显然考虑状态压缩。
- 考虑使用极大极小搜索 **(Minimax)** 结合记忆化求解。
- 提前预处理每个座位的 影响范围状态，可借助位运算将合法性检查优化至 $\mathcal{O}(1)$。

B. Bus 博弈

- 通过递归搜索确定每个状态的最优预期值 $dp[state]$。
- 若当前已落子数为偶数（Alice 回合），取所有后继状态的 max；反之（Bob 回合）取 min。
- 得到 $dp[0]$ 后，从初始态出发，任选满足 $dp[nxt] = dp[curr]$ 的合法转移进行正向推演。
- 迭代至无处落子即为终局，按操作顺序构造 `a/b/x/.` 字符串即可。

时间复杂度： $\mathcal{O}(n \cdot 2^n)$

注：由于数据范围较小，*Alpha-Beta* 剪枝以及蒙特卡洛随机模拟等做法也可以过。

C. 穿越鳌太线

题目要求

在网格中从左上角走到右下角，路径上所有数的按位或结果最小化，在此前提下最大化按位与结果。

考虑从高位到低位贪心。

- 最小化 **OR**：尝试将当前位在 OR 中设为 0。若存在一条路径所有点该位均为 0 则成功，否则该位必须为 1。
- 最大化 **AND**：确定最小 OR 后，在剩余可访问点中再次从高位到低位贪心。尝试将当前位在 AND 中设为 1，若存在一条路径所有点该位均为 1 则成功，否则该位只能为 0。

复杂度：每位 BFS 一次，总复杂度 $\mathcal{O}(31 \cdot nm)$。

L. 龙卷风摧毁停车场

题目大意

$N \times M$ 网格停车场，汽车带朝向（U/D/L/R）。若移动方向无车阻挡则可驶离。求疏散顺序或判定死锁。

- 车辆间的阻挡关系构成有向图。车辆 A 挡住车辆 B，则存在边 $A \to B$。疏散过程实质是求该图的拓扑序。
- $N, M \leq 2000$，车辆数较多。若每次删点后暴力重扫行列维护入度，复杂度将退化至 $\mathcal{O}(K \cdot (N+M))$，无法接受。
- 考虑使用十字链表 $\mathcal{O}(1)$ 维护每个点在四个方向上的最近邻居。

L. 龙卷风摧毁停车场

Step 1: 构建十字链表

扫描网格，预处理出每辆车上、下、左、右第一辆车的坐标。若某车朝向方向的邻居为空（即前方无车），说明其入度为 0，直接压入队列。

Step 2: BFS

车辆出队时，在十字链表中将其 $\mathcal{O}(1)$ 删除。

- 修改该车四个邻居的指针，令其左右邻居互连、上下邻居互连。
- 删点后，仅需检查该点四周的 4 个邻居。若某邻居因前车离开导致其朝向方向变为空，则将其压入队列。

L. 龙卷风摧毁停车场

Step 3: 判环

若最终成功驶离的车辆总数小于初始车辆总数，则说明图中存在环，直接输出 `-1`。

时间复杂度： $\mathcal{O}(NM)$。

J. 树

题目大意

给定一棵 n 个节点的树，初始满足父节点权值 $\geq$ 子节点权值。支持两种操作：

- 查询：寻找节点 x 的祖先中，首个满足权值 $\geq W$ 的节点。
- 修改：将节点 x 权值加 v。若破坏上述单调性则操作失效，否则生效。

操作一：查询

由于根链单调不降，考虑使用树上倍增，在 $\mathcal{O}(\log n)$ 内向上二分跳跃，可以快速定位目标祖先。

J. 树

操作二：修改

若将节点 x 权值修改为 M'_x，仅需校验其是否破坏局部单调性：

$$M_{fa_x} \geq M'_x \geq \max_{y \in son_x} M_y$$

- 为每个节点开一个 `std::multiset` 存储其所有直接子节点的权值。修改生效时同步更新 fa_x 的集合即可。

复杂度：$\mathcal{O}((n+q)\log n)$，同样也可以考虑使用 HLD，常数稍微大一点。

M. 小 t 的 GCD

题目大意

将长度为 n 的数组划分为若干连续子段。子段 $[j,i]$ 贡献为 $(i^2-(j-1)^2)\times \gcd(a_j,\dots,a_i)-P$。求最大总价值。

设 $f[i]$ 表示前 i 个元素划分完毕的最优解。记 $g(j,i)=\gcd(a_j,\dots,a_i)$，朴素转移方程：

$$f[i]=\max_{1\le j\le i}\{f[j-1]-(j-1)^2\cdot g(j,i)+i^2\cdot g(j,i)\}-P$$

提取出与 i 无关（但在同段 gcd 内不变）的项，令

$$V(j)=f[j-1]-(j-1)^2\cdot g(j,i)$$

。

M. 小 t 的 GCD

固定右端点 i，向左延伸的 $\gcd(j,i)$ 会形成至多 $\mathcal{O}(\log(\max A))$ 个值相同的连续段。

我们为这有限的几个连续段，分别维护四个状态元组 $(L,R,v,MaxV)$：

- $[L,R]$ 为该连续段的左右端点。
- v 为该段当前的 gcd 值。
- $MaxV$ 为该段内最大的转移基准值，即：

$$MaxV = \max_{j\in[L,R]}\{f[j-1]-(j-1)^2\cdot v\}$$

M. 小 t 的 GCD

当右端点由 $i-1$ 扩展至 i 时，需令所有段的 $v \leftarrow \gcd(v, a_i)$ 并进行如下维护：

- 若相邻两段更新后的 v 相同，则将它们的区间 $[L, R]$ 合并。
- 仅当某段的 v 发生改变时，由于 $MaxV$ 依赖于 v，我们才暴力遍历 $j \in [L, R]$ 重新计算 $MaxV$。
- 每个位置 j 对应的 gcd 值至多减半 $\log(\max A)$ 次。故暴力重算 $MaxV$ 的均摊总复杂度被严格界定为 $\mathcal{O}(n \log(\max A))$。

M. 小 t 的 GCD

优化后的 DP 转移

对于当前的右端点 i，原本 $\mathcal{O}(n)$ 的合法转移点被浓缩为了 $\mathcal{O}(\log(\max A))$ 个元组。

- 直接遍历当前维护好的连续段集合。
- 对于每一个段 $(L, R, v, MaxV)$，它对 $f[i]$ 的候选贡献为：

$$MaxV + i^2 \cdot v - P$$

- 对所有段的候选贡献取最大值，即为当前的 $f[i]$。

时间复杂度： $\mathcal{O}(n \log(\max A))$
空间复杂度： $\mathcal{O}(n)$

A. Your Shine Your Be!

题目大意

给定非负整数序列，多次在线查询区间 $[l, r]$。要求选出最大且元素互异的子集 M，再从 M 选子集 T，最小化 $|M| \oplus X_T$。

- $|M|$ 的最大值即为区间 $[l, r]$ 中不同元素的个数。
- 由于 M 包含了区间内的所有去重元素，其子集异或和所能张成的线性空间与整个区间 $[l, r]$ 的异或空间完全等价。
- 问题转化为：求常数 $K = |M|$ 与区间 $[l, r]$ 线性基的最小异或和。

A. Your Shine Your Be!

Step 1: 维护区间种类数

考虑一个经典 trick，记录每个元素上一次出现的位置 $last[x]$。一个元素在区间 $[l, r]$ 内首次出现，当且仅当其 $last[x]$ 严格小于 l。

考虑可持久化线段树（主席树）维护该信息：

- 扫描至第 i 个位置时，基于第 $i-1$ 个版本，在位置 $last[a_i]$ 处权值加 1（未出现过则在 0 处加 1）。
- 查询 $[l, r]$ 时，在第 r 个版本的线段树中查询下标区间 $[0, l-1]$ 的权值和，即可在线求出 $K = |M|$。

单次查询时间复杂度 $\mathcal{O}(\log n)$。

A. Your Shine Your Be!

Step 2: 求异或最小值

考虑维护一个带有下标位置 pos 标记的前缀线性基，以处理区间左端点 l 的限制。

- 插入新元素时，若当前位元素的 pos 大于线性基中对应位的 pos，则交换两者，让更靠右的元素留在高位，旧元素继续向低位异或插入。
- 获取 K 后从高到低枚举。若 K 的第 i 位为 1，且前缀线性基该位存在元素且 $pos \geq l$，则令 $K \leftarrow K \oplus basis[i]$，贪心消除高位 1。

时间复杂度： $\mathcal{O}(n \log(\max A) + q(\log n + \log(\max A)))$。

H. 小 t 的递推

题目大意

给定长度为 10^7 的二进制串 s，求 $\sum_{x=1}^{s} f(\text{popcount}(x)) \pmod{10^9+7}$。
其中 f 为递推数列：$f(0)=0, f(1)=1$，$f(i)=a\cdot f(i-1)+f(i-2)$。

- 考虑到 $|s|\le 10^7$，常规数位 DP 若在状态里记录前缀 1 的个数，空间开销会直接导致 MLE。
- 观察发现，当某一位填 0 导致“脱离限制”后，后缀的贡献仅取决于剩余的位数 i。

H. 小 t 的递推

Step 1: 二维 DP

设 $dp[i][j]$ 表示长度为 i 的所有 2^i 种后缀，若其前缀已有 j 个 1，所产生的 f 函数值之和。

即：$dp[i][j] = \sum_{x=0}^{2^i-1} f(\text{popcount}(x) + j)$。

Step 2: 状态转移推导

考虑在长度为 $i-1$ 的后缀前填入 0 或 1。结合 $f(x+2) = a \cdot f(x+1) + f(x)$ 的线性性质，推导如下：

- $dp[i][0] = dp[i-1][0] + dp[i-1][1]$
- $dp[i][1] = dp[i-1][1] + dp[i-1][2]$

  $\Rightarrow dp[i][1] = dp[i-1][1] + (a \cdot dp[i-1][1] + dp[i-1][0])$

  $\Rightarrow dp[i][1] = dp[i-1][0] + (a+1) \cdot dp[i-1][1]$

H. 小 t 的递推

Step 3: 滚动数组降维

发现 j 始终只需维护 0 和 1，且状态仅依赖上一层。故可改变数位 DP 顺序，从右向左倒序扫描 s，直接使用标量滚动维护：

- $dp'_0 = (dp_0 + dp_1) \pmod P$
- $dp'_1 = (dp_0 + (a+1) \cdot dp_1) \pmod P$

每遇到 $s[i] = 1$，直接结合预处理的 f 数组累加贡献即可。

- 时间复杂度： $\mathcal{O}(|s|)$。
- 空间复杂度： $\mathcal{O}(|s|)$。空间瓶颈仅在于存储原串与 f 数组，DP 过程仅占用 $\mathcal{O}(1)$ 辅助空间。

K. MEX

题目大意

给定一棵 n 个节点的树，求包含节点 A 且连通块内权值的 MEX 恰好为 B 的连通块数量。答案对 10^9+7 取模。

Step 1: 容斥

包含 A 且 MEX 为 B 的方案数，等于完全包含 $S(B)\cup\{A\}$ 的方案数，减去完全包含 $S(B+1)\cup\{A\}$ 的方案数。其中 $S(x)$ 表示权值在 $[0,x-1]$ 的节点集合。

K. MEX

Step 2: 引入斯坦纳树

包含点集 V 的最小连通子图即为 V 的斯坦纳树 $T(V)$。连通块的总方案数，就等于 $T(V)$ 所有外向分支的边界乘积。

设 dp_v 表示以 v 为根的子树中，包含 v 的连通块方案数。我们可以得出边界乘积的计算公式：

$$C(V) = \prod_{u \in T(V)} \prod_{\substack{v \in adj(u) \\ v \notin T(V)}} (dp_v + 1)$$

K. MEX

Step 3: 线段树维护一次函数

将查询按 B 离线。通过树链剖分分离轻重儿子，树 DP 的转移方程可重写为：

$$dp_u = (dp_{son} + 1) \prod_{v \in light} (dp_v + 1)$$

令 $k_u = \prod_{v \in light}(dp_v + 1)$，上式即为一次函数 $dp_u = k_u \cdot dp_{son} + k_u$。

每次将新节点接入当前树时，只需利用线段树在重链上动态维护一次函数的复合，并沿轻重链向上更新新的边界乘积即可。

时间复杂度：$\mathcal{O}((n+q)\log^2 n)$

E. 简单数学

题目大意

给定正整数 x，判断是否存在正整数 k，使得 $x \cdot k$ 的十进制表示全为 9。

- 设该全 9 数字长度为 m，其数值可严格表示为 $10^m - 1$。
- 原问题等价于寻找正整数 m，使得 $x \mid (10^m - 1)$。即求解关于 m 的同余方程：

$$10^m \equiv 1 \pmod{x}$$

- 解法：判断 $\gcd(10, x) = 1$ 是否成立即可，以下给出两种证明方式。

E. 简单数学

证法一：欧拉定理

直接考察同余方程 $10^m \equiv 1 \pmod{x}$ 的可解性。

- 充分性：若 $\gcd(10, x) = 1$，由欧拉定理 $10^{\varphi(x)} \equiv 1 \pmod{x}$ 显然成立。直接取 $m = \varphi(x)$ 即为一组合法解。
- 必要性：若 $\gcd(10, x) > 1$，说明 x 含有质因子 2 或 5。而 $10^m - 1$ 尾数恒为 9（必为奇数且不被 5 整除），故绝不可能被含有 2 或 5 的因子整除，方程无解。
- 方程有解的充要条件为 $\gcd(10, x) = 1$，即 x 不含质因子 2 和 5。

E. 简单数学

证法二：抽屉原理

构造全 9 序列，考察其对 x 的取模性质。

- 构造 x 个全 9 的数字：$a_1 = 9, a_2 = 99, \ldots, a_x = \underbrace{99\ldots9}_{x个}$。
- 将这 x 个数对 x 取模，余数范围为 $[0, x-1]$。
- 若均不为 0，由抽屉原理，至少有两数余数相同，设为 a_i, a_j $(i > j)$。
- 作差得：$a_i - a_j = \underbrace{99\ldots9}_{i-j个}\underbrace{00\ldots0}_{j个} = a_{i-j} \times 10^j \equiv 0 \pmod{x}$。
- 即 $x \mid (a_{i-j} \times 10^j)$。要剥离尾部的 0 得到全 9 结果（即使得 $x \mid a_{i-j}$），必须满足 x 与 10^j 互质。结论依然指向 $\gcd(10, x) = 1$。

时间复杂度 $\mathcal{O}(1)$。