

2026 暑期 OI XCPC 交流赛

题解 & Presentation



Rp_Utl & Floris

河南省实验中学

2026 年 7 月 10 日

目录

1 A

2 B

3 C

4 D

5 E

6 F

7 G

8 H

9 I

10 J

11 K

12 L

题目难度分析

预期难度 (Expected)

- **Easy:** $J < A = B < H$
- **Medium:** $I < E < C < D < F$
- **Hard:** $K = G < L$

实际难度 (Actual)

- **Easy:** $J < A < B < I$
- **Medium:** $H < E < C < D < F$
- **Hard:** $K = G < L$

A. 千恋 · 万花

题目大意：在此处直接填入题目大意内容。 n 个物品 (a_i, b_i) ，第一次购买某个物品的花费是 $a_i + b_i$ ，以后再购买这个物品的花费是 b_i 。

初始只能买第 1 个物品，如果要买第 $i (i > 1)$ 个物品，这必须购买过第 $i - 1$ 个物品，求购买 x 个物品的最小花费。

解法

注意到我们购买过的物品构成一个前缀，我们枚举买过的物品前缀 $1 \sim i$ 。

解法

注意到我们购买过的物品构成一个前缀，我们枚举买过的物品前缀 $1 \sim i$ 。

则我们的策略一定是前 i 个物品都买一遍，然后一直购买 $1 \sim i$ 中 b_i 最小的，模拟即可。注意初始化的取值。

时间复杂度： $\mathcal{O}(n)$

B. 你真的掌握了吗?

题目大意:

给定 n 个二元组 (a_i, b_i) , 其中 $b_i \in \{0, 1\}$ 。

确定一个实数 x , 最小化 $\sum_{b_i=0} [a_i \leq x] + \sum_{b_i=1} [a_i \geq x]$ 。

$1 \leq n \leq 2 \times 10^5, 0 \leq b_i \leq 1, 1 \leq a_i \leq 10^9$ 。

解法

不枚举 x , 而是枚举 $b_j = 1$ 的所有二元组中选择的最小的 a_j , 记为 v , 则此时一定会选择所有 $b_j = 1, a_j \geq v$ 的二元组和 $b_j = 0, a_j < v$ 的二元组。

解法

不枚举 x ，而是枚举 $b_j = 1$ 的所有二元组中选择的最小的 a_j ，记为 v ，则此时一定会选择所有 $b_j = 1, a_j \geq v$ 的二元组和 $b_j = 0, a_j < v$ 的二元组。

直接枚举，计算两类二元组的个数可以用二分或者双指针。

时间复杂度： $O(n)/O(n \log n)$ 。

C. 鸟之诗

题目大意：

给你一个 n 个点 m 条边的无向图，边权均为 1，其中有些边的一个端点不确定，记为 k 。

请你对于每一个 $k = 1, 2, 3, \dots, n$ 都求出，此时 $1 \rightarrow n$ 的最短路径。

$$1 \leq n, m \leq 3 \times 10^5。$$

解法

建立一个超级原点 S ，相当于 k 和 S 连一条权值为 0 的边。

解法

建立一个超级原点 S ，相当于 k 和 S 连一条权值为 0 的边。
对于路径不经过超级原点 S 的情况，可以正着跑一遍 BFS 求出来。

解法

建立一个超级原点 S ，相当于 k 和 S 连一条权值为 0 的边。

对于路径不经过超级原点 S 的情况，可以正着跑一遍 BFS 求出来。

否则经过 S ，且路径上一定经过 k ，预处理 f_i 表示 $1 \rightarrow i$ 的最短路径， g_i 表示 $i \rightarrow n$ 的最短路径，答案即为

$\min(f_n, f_k + g_0, f_0 + g_k)$ 。

解法

建立一个超级原点 S ，相当于 k 和 S 连一条权值为 0 的边。

对于路径不经过超级原点 S 的情况，可以正着跑一遍 BFS 求出来。

否则经过 S ，且路径上一定经过 k ，预处理 f_i 表示 $1 \rightarrow i$ 的最短路径， g_i 表示 $i \rightarrow n$ 的最短路径，答案即为

$\min(f_n, f_k + g_0, f_0 + g_k)$ 。

对于 f, g 的处理，可以分别从 $1, n$ 为起点跑两遍 BFS。

时间复杂度： $O(n)$ 。

D. 愿望实现的小镇

题目大意：

给你一个长度为 n 的字符串 S 和一个长度为 m 的字符串 T 。
请你最小化 k ，使得 T 可以被分割成 k 个串，每个串都是 S 的前缀。

$$1 \leq n, m \leq 5 \times 10^5。$$

解法

首先, 对于每个 i , 可以求出 $T_{i \sim m}$ 和 S 匹配的最长前缀长度, 记为 r_i 。

解法

首先, 对于每个 i , 可以求出 $T_{i \sim m}$ 和 S 匹配的最长前缀长度, 记为 r_i 。

则问题可以转化为, 有 n 个区间 $[i, r_i]$, 求覆盖 $1 \sim n$ 的所有整点, 最少选择的区间个数。

解法

首先, 对于每个 i , 可以求出 $T_{i \sim m}$ 和 S 匹配的最长前缀长度, 记为 r_i 。

则问题可以转化为, 有 n 个区间 $[i, r_i]$, 求覆盖 $1 \sim n$ 的所有整点, 最少选择的区间个数。

这是一个经典的问题, 可以数据结构优化 dp, 也可以直接贪心, 复杂度为 $O(n \log n)$, 后者更容易实现。

解法

首先，对于每个 i ，可以求出 $T_{i \sim m}$ 和 S 匹配的最长前缀长度，记为 r_i 。

则问题可以转化为，有 n 个区间 $[i, r_i]$ ，求覆盖 $1 \sim n$ 的所有整点，最少选择的区间个数。

这是一个经典的问题，可以数据结构优化 dp，也可以直接贪心，复杂度为 $O(n \log n)$ ，后者更容易实现。

本题存在一个更好写的 $O(n)$ 的 KMP 解法，感兴趣的选手可以自行思考。

时间复杂度： $O(n \log n) / O(n)$ 。

E. 何为容器?

题目大意:

给你一个有 n 个点的环, 每个环上的点有一个权值 $w_i (0 \leq i < n)$ 。

从第 0 个点开始, 维护变量 s, r , 初始为 0, 按照 $0 \rightarrow n \rightarrow 0 \rightarrow \dots$ 的顺序依次执行 $s \leftarrow s + w_i, r \leftarrow r + 1$, 如果 s 超过给定的阈值 x , 就输出 r , 并将 s, r 清空。

q 次询问, 每次询问第 k 个输出的数。

$1 \leq n, q \leq 2 \times 10^5, 1 \leq w_i \leq 10^9, 1 \leq x \leq 10^9, k \leq 10^{12}$ 。

解法

首先可以想预处理出所有的 r ，但是太大了，不过不难发现经过一段时间后， r 会出现循环节。

解法

首先可以想预处理出所有的 r ，但是太大了，不过不难发现经过一段时间后， r 会出现循环节。

可以预处理出从第 i 个点开始取，到哪个位置会超过阈值，找循环节即可。对于预处理，可以前缀和加二分实现。

时间复杂度： $O(n \log n + q)$ 。

F. 我玩蔚蓝从不红温！

题目大意：给你一个二维平面，你可以从 (x, y) 移动到 $(x-1, y), (x+1, y), (x, y-1), (x, y+1)$ 。

对于一次移动，如果起点和终点都在 $x = Bn$ 或者 $y = Bn$ 的直线上（这样的直线上成为大道），则这一次移动的代价是 1，否则是 k 。求从 (a, b) 出发移动到 (u, v) 的最小代价。

T 次询问，每次给出 k, B, a, b, u, v 求解。

$1 \leq T \leq 2 \times 10^5, 1 \leq b, k \leq 10^9, 0 \leq a, b, u, v \leq 10^9$ 。

解法

首先可以直接沿着曼哈顿距离最小的路径走，每一步代价为 k 。

解法

首先可以直接沿着曼哈顿距离最小的路径走，每一步代价为 k 。
其次，我们可以先让两个点移动到一个方向的一个大道上（每一步花费 k ），然后沿着大道移动到大道的交叉点上，最后沿着大道相遇（每一步花费为 1）。

解法

首先可以直接沿着曼哈顿距离最小的路径走，每一步代价为 k 。
其次，我们可以先让两个点移动到一个方向的一个大道上（每一步花费 k ），然后沿着大道移动到大道的交叉点上，最后沿着大道相遇（每一步花费为 1）。

实际上，如果两个点移动到一个方向的同一个大道上，可以直接相遇。所有的情况都可以被这三种情况归纳，模拟即可。可能比较难写。

时间复杂度： $O(T)$ ，但是大常数。

G. 杳見玛——杀

题目大意：

给定 n, S 和数列 $a_{1 \sim n}$ ，计算符合条件的数列个数，条件如下：

1. 序列和为 S 。
2. 序列的前缀和中不能出现 $a_{1 \sim n}$ 的任何一种元素。

解法

考虑 dp, 设 f_i 表示和为 i 的子序列个数, 枚举最后一个数的取值, 有 $f_i \sum_j f_{i-2j+1}$ 的转移式。

解法

考虑 dp, 设 f_i 表示和为 i 的子序列个数, 枚举最后一个数的取值, 有 $f_i \sum_j f_{i-2j+1}$ 的转移式。
这样做有一个非常好的优势, 就是在递推过程中可以把 f_{a_i} 的项清零, 这样就处理了限制。

解法

考虑 dp, 设 f_i 表示和为 i 的子序列个数, 枚举最后一个数的取值, 有 $f_i \sum_j f_{i-2j+1}$ 的转移式。

这样做有一个非常好的优势, 就是在递推过程中可以把 f_{a_i} 的项清零, 这样就处理了限制。

直接递推的复杂度为 $O(V)$, 显然会爆, 不过我们发现这个转移式非常的简单, f 会和所有下标奇偶性和他不同的项转移。

解法

考虑 dp, 设 f_i 表示和为 i 的子序列个数, 枚举最后一个数的取值, 有 $f_i \sum_j f_{i-2j+1}$ 的转移式。

这样做有一个非常好的优势, 就是在递推过程中可以把 f_{a_i} 的项清零, 这样就处理了限制。

直接递推的复杂度为 $O(V)$, 显然会爆, 不过我们发现这个转移式非常的简单, f 会和所有下标奇偶性和他不同的项转移。

考虑用矩阵乘法来描述 dp 的转移, 设 $F_i = [f_i \ g_1 \ g_0]$, 其中 $g_k = \sum_{j \bmod 2=k} f_j (j < i)$ 。

解法

当 i 为偶数时，转移矩阵为：

$$G_0 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

解法

当 i 为偶数时，转移矩阵为：

$$G_0 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

当 i 为奇数时，转移矩阵为：

$$G_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

解法

初始有 $F_i = [0 \ 0 \ 1]$, 我们有 $F_i \times G_{i \bmod 2} = F_{i+1}$ 。

解法

初始有 $F_i = [0 \ 0 \ 1]$, 我们有 $F_i \times G_{i \bmod 2} = F_{i+1}$ 。

我们只需要在相邻的 $[a_i, a_{i+1})$ 区域内做矩阵快速幂加速转移, 在特殊节点处清空 dp 值即可。预处理出 $G_0 G_1$ 的次幂, 处理一些小细节即可。

时间复杂度: 预处理 $O(k^3 \log V)$, dp 过程可以 $O(nk^2 \log n)$, 其中 $k = 3$ 。

H.Mujica 压中高考？这你不看？！

题目大意：

我们需要确定一个最小的 S ，使得存在一个地点，从该点出发能到达所有的点

移动时需满足 $P_i \times S \geq |x_i - x_j| + |y_i - y_j|$ 。

$1 \leq N \leq 200, 1 \leq P_i \leq 10^9$

解法

做法一: 二分 + dfs

解法

做法一: 二分 + dfs

我们首先发现这个题面实际是有单调性的, 如果 S 满足, 那么 $S + 1$ 必然满足。

解法

做法一：二分 + dfs

我们首先发现这个题面实际是有单调性的，如果 S 满足，那么 $S+1$ 必然满足。

那么我们直接二分答案，对于每一个点 i 直接 dfs 求连通块，看看是否有大小为 n 的连通块。

时间复杂度： $\mathcal{O}(n^3 \log S)$

做法二：FLloyd

解法

做法一：二分 + dfs

我们首先发现这个题面实际是有单调性的，如果 S 满足，那么 $S+1$ 必然满足。

那么我们直接二分答案，对于每一个点 i 直接 dfs 求连通块，看看是否有大小为 n 的连通块。

时间复杂度： $\mathcal{O}(n^3 \log S)$

做法二：FLloyd

我们实际上发现这个过程很类似于传递闭包，我们如果能知道 i 到 j 的最小训练次数就好了。

解法

做法一：二分 + dfs

我们首先发现这个题面实际是有单调性的，如果 S 满足，那么 $S+1$ 必然满足。

那么我们直接二分答案，对于每一个点 i 直接 dfs 求连通块，看看是否有大小为 n 的连通块。

时间复杂度： $\mathcal{O}(n^3 \log S)$

做法二：FLoyd

我们实际上发现这个过程很类似于传递闭包，我们如果能知道 i 到 j 的最小训练次数就好了。

那么我们是否可以直接可以使用一个广为人知的算法 Floyd，做出上面的操作之后，直接对于每个点取 $\max$ ，最终取的是 $\min(\max ans_i)$ 。

时间复杂度： $\mathcal{O}(n^3)$

I. 白色相簿的季节又到了

题目大意：

我们的总金额是 N ，每次操作可以添加一个数位，每个数 $i (i \in [1, 9])$ 都有花费 C_i ，求最大的能拼出的数是啥
 $C_i \leq N, 1 \leq N \leq 10^6$

解法

注意到数位越多越优。

解法

注意到数位越多越优。

那么我们的做法就很显然，确定数位的最大 $Length$ ，确定之后看剩余的花费 $Cost$ ，再由最高位向最低位替换，替换的时候保证先换最大的一定最优。

时间复杂度： $\mathcal{O}(9N)$

J. 有艾拉我就来打比赛了

题目大意：

有 K 个棋子，初始位置是严格递增的，每个格子最多一个棋子，且棋子之间的相对顺序不能改变。

每次操作：给定一个棋子索引 L （从左到右数第 L 个棋子）如果它已经在最右边（位置 N ），则不动否则，如果它右边的格子为空，就向右移动一格否则不动

$1 \leq K, N \leq 200, 1 \leq Q \leq 1000$

解法

简单模拟即可。

K. 最幸福的女孩

题目大意：每个格子只能被一个斜率为 2 且有宽度限制的格子覆盖，问覆盖的部分有多少个格子是有棋子的。

$$1 \leq N \leq 2000$$

解法

考虑二维的差分（或者说是容斥）。

解法

考虑二维的差分（或者说是容斥）。

先把规则转换之后考虑如何做。

我们考虑一个神奇的差分方式，每次都做到从 $1 \sim i$ 行，第 i 行从 $i \sim 1$ 行每行的数量分别为 $j + (i - \text{now}) * 2$ ，这样我们可以差分出一个很好的区域。

经过差分我们的查询将可以做到 $\mathcal{O}(1)$ 了。

时间复杂度： $\mathcal{O}(n^2)$ 。

L. 提瓦特实验中学

题目大意：

给定一棵有 N 个顶点的树。求出最小的 K 使得对于每一种颜色，被该颜色染色的顶点集合在树中是连通的，并且构成一条简单路径。对于树上的任意一条简单路径，该路径上包含的不同颜色的种类数不超过 K 。

$$1 \leq n \leq 2 \times 10^5$$

解法

我们注意到这个 K 还是有单调性的，因为对于 K 成立，那么 $K+1$ 一定是成立的。并且对于一棵树来说，剖分为若干条重链之后，从根走到叶子节点经过的点最多有 $\log n$ 个。

解法

我们注意到这个 K 还是有单调性的，因为对于 K 成立，那么 $K+1$ 一定是成立的。并且对于一棵树来说，剖分为若干条重链之后，从根走到叶子节点经过的点最多有 $\log n$ 个。
考虑二分答案，接着考虑如何 check。

解法

我们注意到这个 K 还是有单调性的，因为对于 K 成立，那么 $K+1$ 一定是成立的。并且对于一棵树来说，剖分为若干条重链之后，从根走到叶子节点经过的点最多有 $\log n$ 个。

考虑二分答案，接着考虑如何 check。

当前的颜色只和子树内的情况应当是有关系的，我们发现对于点 u 来说，有以下情况：

3. u 作为一条链的链尾，即子树内颜色与 u 不一样，我们定义为 $f_{u,0}$ 。

4. u 作为上面链的延申，与子树内颜色至多有一个链一样的，我们定义为 $f_{u,1}$ 。

5. u 作为一个链的链首，与子树内可以有两条链，一个折线形，定义为 $f_{u,2}$ 。

初始的 $f_{u,i} = 1$ 表示只有自己这个颜色段。 $f_{u,i}$ 表示 u 这个点到叶子节点最小可能的最大值。

解法

然后我们考虑 u 和 v 的转移，对于 u 和 v 的颜色进行分类讨论。

6. u 和 v 不同色时，我们只需要保证 $f_{u,i} + f_{v,2} \leq k$ 的时候我们

$g_i \rightarrow \min(g_i, \max(f_{u,i}, f_{v,2} + 1))$

7. u 和 v 同色时，我们需要保证 $i < 2$ 并且 $f_{u,i} + f_{v,1} - 1 \leq k$ 时，

我们就可以让 v 接到 u 的后面，让

$g_{i+1} = \min(g_{i+1}, \max(f_{u,i}, f_{v,1}))$ 。

转移的时候注意到 g_i 的约束是至少，所以 g_i 能从比 i 小的转移。

最后的 check 只需要判断 $f_{1,2}$ 与 mid 的大小关系进行更新即可。

时间复杂度： $\mathcal{O}(n \log \log n)$

Thank You!

感谢大家的聆听与参与