The 2021 Hangzhou Normal U Summer Trials
Tutorial

Hangzhou Normal U ACM/ICPC Team

2021 年 05 月 29 日

1. Solution

Problem A Aahaxiki's journey I - set off

- 同城内任意场地的转移需要额外的时间金钱花费

Problem A Aahaxiki's journey I - set off

- 同城内任意场地的转移需要额外的时间金钱花费
- A 火车站 - 火车 → B 火车站 - 火车 → C 火车站

Problem A Aahaxiki's journey I - set off

- 同城内任意场地的转移需要额外的时间金钱花费
- A 火车站 - 火车 $\rightarrow$ B 火车站 - 火车 $\rightarrow$ C 火车站
- A 火车站 - 火车 $\rightarrow$ B 火车站 - 额外花费 $\rightarrow$ B 机场 - 飞机 $\rightarrow$ C 机场

Problem A Aahaxiki's journey I - set off

- 同城内任意场地的转移需要额外的时间金钱花费
- A 火车站 - 火车 $\rightarrow$ B 火车站 - 火车 $\rightarrow$ C 火车站
- A 火车站 - 火车 $\rightarrow$ B 火车站 - 额外花费 $\rightarrow$ B 机场 - 飞机 $\rightarrow$ C 机场
- 考虑拆点，将每个城市的每个场地视为一个独立的点，每个城市里任意两个场地建无向边

Problem A Aahaxiki's journey I - set off

- 同城内任意场地的转移需要额外的时间金钱花费
- A 火车站 - 火车 $\rightarrow$ B 火车站 - 火车 $\rightarrow$ C 火车站
- A 火车站 - 火车 $\rightarrow$ B 火车站 - 额外花费 $\rightarrow$ B 机场 - 飞机 $\rightarrow$ C 机场
- 考虑拆点，将每个城市的每个场地视为一个独立的点，每个城市里任意两个场地建无向边
- $T(1 \leq T \leq 10^5)$ 组输入，单组输入最多 $N = 10^5$ 个点，注意初始化

Problem A Aahaxiki's journey I - set off

- 同城内任意场地的转移需要额外的时间金钱花费
- A 火车站 - 火车 $\rightarrow$ B 火车站 - 火车 $\rightarrow$ C 火车站
- A 火车站 - 火车 $\rightarrow$ B 火车站 - 额外花费 $\rightarrow$ B 机场 - 飞机 $\rightarrow$ C 机场
- 考虑拆点，将每个城市的每个场地视为一个独立的点，每个城市里任意两个场地建无向边
- $T(1 \leq T \leq 10^5)$ 组输入，单组输入最多 $N = 10^5$ 个点，注意初始化
- memset(dis, 0, sizeof dis) - int dis[N] - TLE

Problem A Aahaxiki's journey I - set off

- 同城内任意场地的转移需要额外的时间金钱花费
- A 火车站 - 火车 $\rightarrow$ B 火车站 - 火车 $\rightarrow$ C 火车站
- A 火车站 - 火车 $\rightarrow$ B 火车站 - 额外花费 $\rightarrow$ B 机场 - 飞机 $\rightarrow$ C 机场
- 考虑拆点，将每个城市的每个场地视为一个独立的点，每个城市里任意两个场地建无向边
- $T(1 \leq T \leq 10^5)$ 组输入，单组输入最多 $N = 10^5$ 个点，注意初始化
- memset(dis, 0, sizeof dis) - int dis[N] - TLE
- 考虑 (n == 1)，规定本校和比赛场地不在同一位置，因此需要花费 x、y 去往

Problem B Bsueh- and Gold Medals

- 没有比较好的贪心策略

Problem B Bsueh- and Gold Medals

- 没有比较好的贪心策略
- 至少有两块金牌，所以 p 的最大值是确定的

Problem B Bsueh- and Gold Medals

- 没有比较好的贪心策略
- 至少有两块金牌，所以 p 的最大值是确定的
- 那么直接二分，每次贪心地把尽量大的放在下面，进行 check 即可

Problem C Chtholly and Floating Islands

- 典型算路径数的 *dp* 问题

Problem C Chtholly and Floating Islands

- 典型算路径数的 dp 问题
- 存在不能走的情况，数据范围很小，能用状压处理

Problem C Chtholly and Floating Islands

- 典型算路径数的 dp 问题
- 存在不能走的情况，数据范围很小，能用状压处理
- q 较多，每次询问都跑一遍会超时。

Problem C Chtholly and Floating Islands

- 典型算路径数的 dp 问题
- 存在不能走的情况，数据范围很小，能用状压处理
- q 较多，每次询问都跑一遍会超时。
- 已知所有情况，可以提前预处理出答案，$O(1)$ 查询。

Problem D Dllllan and his friends

- 对于 $lllllan's hourse$ 我们任取三点构造一个圆，判断其他点是否在圆上。

Problem D DIllllan and his friends

- 对于 *lllllan′shourse* 我们任取三点构造一个圆，判断其他点是否在圆上。
- 如果有点不在圆上，则找不到 *lllllan′s hourse*

Problem D Dllllan and his friends

- 对于 *lllllan′shourse* 我们任取三点构造一个圆，判断其他点是否在圆上。
- 如果有点不在圆上，则找不到 *lllllan′s hourse*
- 若所有点都在圆上，则构建一棵 *MST*

Problem E Ewo Slices of Bread with Cheese

- 很明显的贪心做法是从偶数开始，奇偶轮流取

Problem E Ewo Slices of Bread with Cheese

- 很明显的贪心做法是从偶数开始，奇偶轮流取
- 同奇同偶不能连续取，最优情况下取完的天数是确定的

Problem E Ewo Slices of Bread with Cheese

- 很明显的贪心做法是从偶数开始，奇偶轮流取
- 同奇同偶不能连续取，最优情况下取完的天数是确定的
- 算出天数取大即可

Problem F Fstee1XD and Minioins

- 假设当前是第 i 天，根据题意第 $i-1$ 天对第 i 天贡献了 $2f[i-1]$; 前 $i-2$ 天对第 i 天贡献了 $\sum_{j=1}^{i-2} f[j]$。故

Problem F Fstee1XD and Minioins

- 假设当前是第 i 天，根据题意第 $i-1$ 天对第 i 天贡献了 $2f[i-1]$; 前 $i-2$ 天对第 i 天贡献了 $\sum_{j=1}^{i-2} f[j]$。故
-

$$
\begin{aligned}
& f[i] && (1) \\
=& 2 * f[i-1] + \sum_{j=1}^{i-2} f[j] && (2) \\
=& 2 * f[i-1] + f[i-1] - f[i-2] && (3) \\
=& 3 * f[i-1] - f[i-2] && (4)
\end{aligned}
$$

Problem F Fstee1XD and Minioins

- 由于 n 很大，故考虑矩阵快速幂，构造矩阵 A 满足

Problem F Fstee1XD and Minioins

- 由于 n 很大，故考虑矩阵快速幂，构造矩阵 A 满足
- $$\begin{bmatrix} f_{n-1} \\ f_{n-2} \end{bmatrix} * A = \begin{bmatrix} f_n \\ f_{n-1} \end{bmatrix}$$

Problem F Fstee1XD and Minioins

- 由于 n 很大，故考虑矩阵快速幂，构造矩阵 A 满足
- $$\begin{bmatrix} f_{n-1} \\ f_{n-2} \end{bmatrix} * A = \begin{bmatrix} f_n \\ f_{n-1} \end{bmatrix}$$
- $$A = \begin{bmatrix} 3 & -1 \\ 1 & 0 \end{bmatrix}$$

Problem G Guess Permutation

- 数据范围给了二进制做法的提示

Problem G Guess Permutation

- 数据范围给了二进制做法的提示
- 每个数字都是不一样的，那么我们对所有数字从 1 到 n 进行编号

Problem G Guess Permutation

- 数据范围给了二进制做法的提示
- 每个数字都是不一样的，那么我们对所有数字从 1 到 n 进行编号
- 将编号作为 2 进制表示，第 i 次询问所有在第 $(i-1)$ 位为 1 的编号，计算贡献，就能得到最终的答案。

Problem H Hsueh- and Meeting

- 考虑将一个人和一个座位绑定

Problem H Hsueh- and Meeting

- 考虑将一个人和一个座位绑定
- 先选一个位置给第一个人坐下

Problem H Hsueh- and Meeting

- 考虑将一个人和一个座位绑定
- 先选一个位置给第一个人坐下
- 再选剩下的人，剩下还有 $m-2-(n-1)$ 个位置，选 $n-1$ 个位置坐人

Problem H Hsueh- and Meeting

- 考虑将一个人和一个座位绑定
- 先选一个位置给第一个人坐下
- 再选剩下的人，剩下还有 $m-2-(n-1)$ 个位置，选 $n-1$ 个位置坐人
- 因为人是不同的，故剩下 $n-1$ 个人需要全排列

Problem H Hsueh- and Meeting

- 考虑将一个人和一个座位绑定
- 先选一个位置给第一个人坐下
- 再选剩下的人，剩下还有 $m-2-(n-1)$ 个位置，选 $n-1$ 个位置坐人
- 因为人是不同的，故剩下 $n-1$ 个人需要全排列
- 特判只有一个人的情况

Problem I Iahaxiki's journey II - enjoying

- 要求从树上找一个含 k 个节点的最长路

Problem I Iahaxiki's journey II - enjoying

- 要求从树上找一个含 k 个节点的最长路
- 实在不会，考虑最笨做法

Problem I Iahaxiki's journey II - enjoying

- 要求从树上找一个含 k 个节点的最长路
- 实在不会，考虑最笨做法
- 从每个点出发，树上 DFS k 个深度，求最长路

Problem I Iahaxiki's journey II - enjoying

- 要求从树上找一个含 k 个节点的最长路
- 实在不会，考虑最笨做法
- 从每个点出发，树上 DFS k 个深度，求最长路
- 考虑极端情况的复杂度 $O(n * d)$(d - 树的直径)

Problem I Iahaxiki's journey II - enjoying

- 要求从树上找一个含 k 个节点的最长路
- 实在不会，考虑最笨做法
- 从每个点出发，树上 DFS k 个深度，求最长路
- 考虑极端情况的复杂度 $O(n * d)$(d - 树的直径)
- 菊花图 $O(n * 3)$

Problem I Iahaxiki's journey II - enjoying

- 要求从树上找一个含 k 个节点的最长路
- 实在不会，考虑最笨做法
- 从每个点出发，树上 DFS k 个深度，求最长路
- 考虑极端情况的复杂度 $O(n*d)$(d - 树的直径)
- 菊花图 $O(n*3)$
- 链 $O(n*n)$

Problem I Iahaxiki's journey II - enjoying

- 要求从树上找一个含 k 个节点的最长路
- 实在不会，考虑最笨做法
- 从每个点出发，树上 DFS k 个深度，求最长路
- 考虑极端情况的复杂度 $O(n*d)$(d - 树的直径)
- 菊花图 $O(n*3)$
- 链 $O(n*n)$
- 输入保证 T 组输入 n 的和不超过 10^6，最坏情况复杂度近似为 $O(10*\sum_{i=1}^{N} i)$

Problem I Iahaxiki's journey II - enjoying

- 进一步考虑，树形 *DP* 减少一些重复计算

Problem I Iahaxiki's journey II - enjoying

- 进一步考虑，树形 DP 减少一些重复计算
- 剖出每个点为根的子树，考虑子树上经过根节点最长路

Problem I Iahaxiki's journey II - enjoying

- 进一步考虑，树形 *DP* 减少一些重复计算
- 剖出每个点为根的子树，考虑子树上经过根节点最长路
- 定义 $dp[len]$ 记录从根节点 rt 出发 len 个长度的最长路，ans = max{dp[i] + dp[k - i]}

Problem I Iahaxiki's journey II - enjoying

- 进一步考虑，树形 DP 减少一些重复计算
- 剖出每个点为根的子树，考虑子树上经过根节点最长路
- 定义 $dp[len]$ 记录从根节点 rt 出发 len 个长度的最长路，ans = max{dp[i] + dp[k - i]}
- 考虑极端情况的复杂度 $O(\sum_{i=1}^{n} d_i)(d_i$ - 子树的直径$)$

Problem I Iahaxiki's journey II - enjoying

- 进一步考虑，树形 DP 减少一些重复计算
- 剖出每个点为根的子树，考虑子树上经过根节点最长路
- 定义 $dp[len]$ 记录从根节点 rt 出发 len 个长度的最长路，ans = max{dp[i] + dp[k - i]}
- 考虑极端情况的复杂度 $O(\sum_{i=1}^{n} d_i)(d_i$ - 子树的直径)
- 菊花图 $O(n)$

Problem I Iahaxiki's journey II - enjoying

- 进一步考虑，树形 DP 减少一些重复计算
- 剖出每个点为根的子树，考虑子树上经过根节点最长路
- 定义 $dp[len]$ 记录从根节点 rt 出发 len 个长度的最长路，ans = max{dp[i] + dp[k - i]}
- 考虑极端情况的复杂度 $O(\sum_{i=1}^{n} d_i)(d_i$ - 子树的直径)
- 菊花图 $O(n)$
- 链 $O(\sum_{i=1}^{n} i)$

Problem I Iahaxiki's journey II - enjoying

- 进一步考虑，树形 DP 减少一些重复计算
- 剖出每个点为根的子树，考虑子树上经过根节点最长路
- 定义 $dp[len]$ 记录从根节点 rt 出发 len 个长度的最长路，ans = max{dp[i] + dp[k - i]}
- 考虑极端情况的复杂度 $O(\sum_{i=1}^{n} d_i)(d_i$ - 子树的直径)
- 菊花图 $O(n)$
- 链 $O(\sum_{i=1}^{n} i)$
- 输入保证 T 组输入 n 的和不超过 10^6，最坏情况复杂度近似为 $O(10 * \sum_{i=1}^{n} i)$

Problem I Iahaxiki's journey II - enjoying

- 点分治

Problem I Iahaxiki's journey II - enjoying

- 点分治
- 两个办法都是因为极端情况-链，弄得复杂度极高

Problem I lahaxiki's journey II - enjoying

- 点分治
- 两个办法都是因为极端情况-链，弄得复杂度极高
- 沿用动态规划的思路，考虑找每课子树的重心以减小树直径的长度

Problem I Iahaxiki's journey II - enjoying

- 点分治
- 两个办法都是因为极端情况-链，弄得复杂度极高
- 沿用动态规划的思路，考虑找每课子树的重心以减小树直径的长度
- 考虑极端情况的复杂度 $O(\sum_{i=1}^{n} d_i)(d_i$ - 子树的直径)

Problem I lahaxiki's journey II - enjoying

- 点分治
- 两个办法都是因为极端情况-链，弄得复杂度极高
- 沿用动态规划的思路，考虑找每课子树的重心以减小树直径的长度
- 考虑极端情况的复杂度 $O(\sum_{i=1}^{n} d_i)(d_i$ - 子树的直径$)$
- 菊花图 $O(n)$

Problem I Iahaxiki's journey II - enjoying

- 点分治
- 两个办法都是因为极端情况-链，弄得复杂度极高
- 沿用动态规划的思路，考虑找每课子树的重心以减小树直径的长度
- 考虑极端情况的复杂度 $O(\sum_{i=1}^{n} d_i)(d_i$ - 子树的直径)
- 菊花图 $O(n)$
- 链 $O(n \log n)$

Problem I Iahaxiki's journey II - enjoying

- 点分治
- 两个办法都是因为极端情况-链，弄得复杂度极高
- 沿用动态规划的思路，考虑找每课子树的重心以减小树直径的长度
- 考虑极端情况的复杂度 $O(\sum_{i=1}^{n} d_i)(d_i$ - 子树的直径)
- 菊花图 $O(n)$
- 链 $O(n \log n)$
- 输入保证 T 组输入 n 的和不超过 10^6，最坏情况复杂度近似为 $O(10 * n \log n)$

Problem J Jahaxiki's journey III - Tryna lost

- 输入是一个二维平面的示意图，每条线视为一条边，四条边围成一个最小单位。两个最小单位之间没有线表示两个单位连通。问图中是否有存在环

Problem J Jahaxiki's journey III - Tryna lost

- 输入是一个二维平面的示意图，每条线视为一条边，四条边围成一个最小单位。两个最小单位之间没有线表示两个单位连通。问图中是否有存在环
- 考虑用并查集判环

Problem J Jahaxiki's journey III - Tryna lost

- 输入是一个二维平面的示意图，每条线视为一条边，四条边围成一个最小单位。两个最小单位之间没有线表示两个单位连通。问图中是否有存在环
- 考虑用并查集判环
- 观察输入格式，遍历每个最小单位，考虑是否与右方和下方单位合并即可。

Problem J Jahaxiki's journey III - Tryna lost

- 输入是一个二维平面的示意图，每条线视为一条边，四条边围成一个最小单位。两个最小单位之间没有线表示两个单位连通。问图中是否有存在环
- 考虑用并查集判环
- 观察输入格式，遍历每个最小单位，考虑是否与右方和下方单位合并即可。
- 也可以用 *DFS*

Thank You!