JOISC 原题典题不可做题选讲

ningago

兰州树人中学

2023 年 7 月 14 日

「JOISC 2022 Day1」京都观光

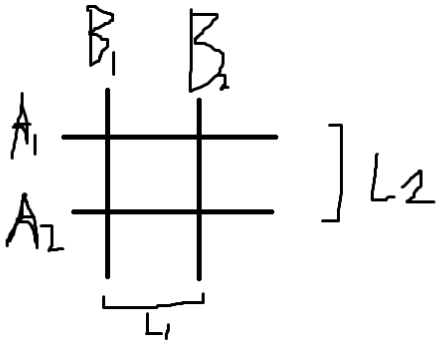
题意简述：

城市布局为 $n \times m$ 的网格图。在第 i 条横向路径上行走一单位长度的时间是 A_i，在第 i 条纵向路径上行走一单位长度的时间是 B_i。

请求出从 $(1,1)$ 到达 (n,m) 的最短时间。

$n, m \leq 10^5, A_i, B_i \leq 10^9$。

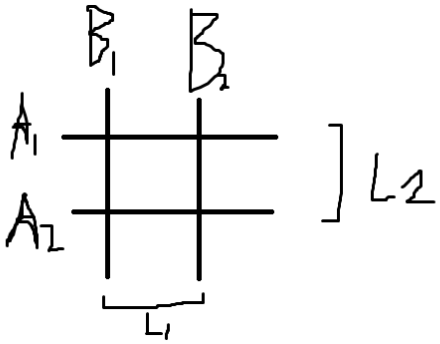
「JOISC 2022 Day1」京都观光



考虑 $A_1 \to B_2$ 优于 $B_1 \to A_2$ 的条件：

（L 是指两道路之间的距离，初始时所有 $\Delta L = 1$，随着删边的过程增大。）

「JOISC 2022 Day1」京都观光



考虑 $A_1 \to B_2$ 优于 $B_1 \to A_2$ 的条件：

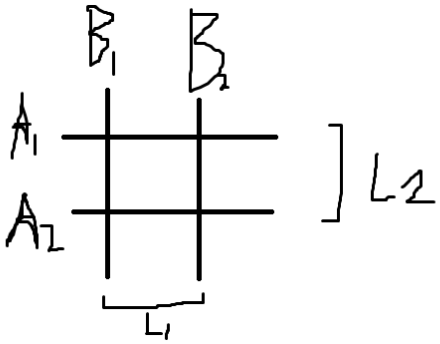
（L 是指两道路之间的距离，初始时所有 $\Delta L = 1$，随着删边的过程增大。）

$$A_1 \cdot L_1 + B_2 \cdot L_2 \geq A_2 \cdot L_1 + B_1 \cdot L_2$$

$$\Longrightarrow \frac{A_1 - A_2}{L_2} \geq \frac{B_1 - B_2}{L_2}$$

$$\Longrightarrow \frac{A_2 - A_1}{L_2} \leq \frac{B_2 - B_1}{L_2}$$

「JOISC 2022 Day1」京都观光



考虑 $A_1 \to B_2$ 优于 $B_1 \to A_2$ 的条件：

（L 是指两道路之间的距离，初始时所有 $\Delta L = 1$，随着删边的过程增大。）

$$A_1 \cdot L_1 + B_2 \cdot L_2 \geq A_2 \cdot L_1 + B_1 \cdot L_2$$

$$\Longrightarrow \frac{A_1 - A_2}{L_2} \geq \frac{B_1 - B_2}{L_2}$$

$$\Longrightarrow \frac{A_2 - A_1}{L_2} \leq \frac{B_2 - B_1}{L_2}$$

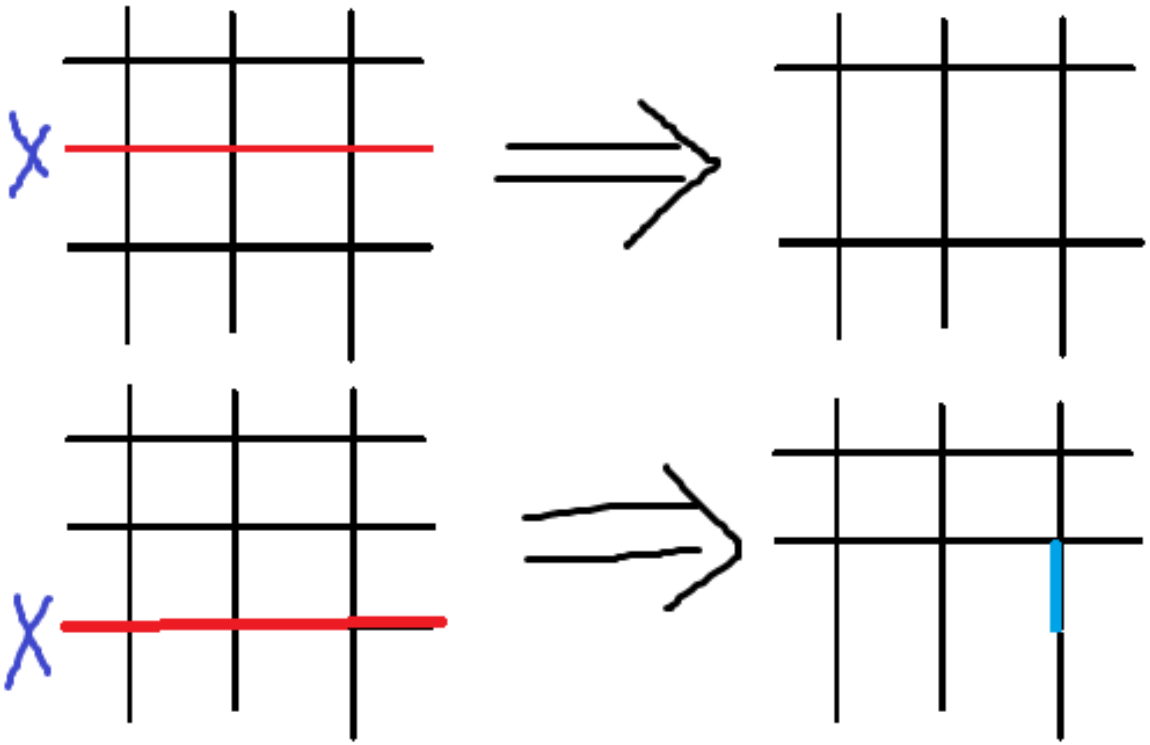
所以 $\dfrac{\Delta A_i / \Delta B_i}{\Delta L_i}$ 最大的道路必然劣于其他道路。

所以策略为：每次选择 $\dfrac{\Delta A_i / \Delta B_i}{\Delta L_i}$ 最大的道路，将其删去。

如果其为最后一条纵/横向边，则统计其对答案的贡献。

「JOISC 2022 Day1」京都观光

图示：（红色表示删去边，蓝色表示计入答案）



形象化：https://www2.ioi-jp.org/camp/2022/2022-sp-tasks/contest1/kyoto-review.pdf#page=33&zoom=auto,-85,540。

「JOISC 2022 Day4」复兴计划

题意简述：

给定带权无向联通图，n 个点 m 条边。每条边初始权值为 val_i。

询问 Q 次，每次给定 q_i，求每条边 j 的权值变为 $|q_i - val_j|$ 时的最小生成树。

$n \leq 500, m \leq 10^5, Q \leq 10^6$，$q_i$ 单调递增。

「JOISC 2022 Day4」复兴计划

首先根据 kruskal 的过程易得最小生成树具有可合并性
即令 $MST(E) \subseteq E$ 表示构成 E（边集）的最小生成树的边集，则有：

$$MST(E1 \cup E2) \subseteq MST(E1) \cup MST(E2)$$

「JOISC 2022 Day4」复兴计划

首先根据 kruskal 的过程易得最小生成树具有可合并性
即令 $MST(E) \subseteq E$ 表示构成 E（边集）的最小生成树的边集，则有：

$$MST(E1 \cup E2) \subseteq MST(E1) \cup MST(E2)$$

所以现在可以口糊一个算法：动态维护边权 $\leq q_i$ 和 $> q_i$ 的两个边集 E_L, E_R，每次 q_i 的增大相当于从 E_R 中拿出一些边到 E_L 中去，然后求 $E_L \cup E_R$ 的 MST。

考虑绝对值函数 $f(x) = |x - a|$ 的性质。

「JOISC 2022 Day4」复兴计划

考虑绝对值函数 $f(x)=|x-a|$ 的性质。

由于斜率对于所有的 a 来说都恒定为 -1 和 1，所以对于两个绝对值函数 $f_1,f_2(a_1\le a_2)$ 来说：

若 $x<\dfrac{a_1+a_2}{2}$，则 $f_1(x)>f_2(x)$，否则 $f_1(x)<f_2(x)$。

「JOISC 2022 Day4」复兴计划

考虑绝对值函数 $f(x)=|x-a|$ 的性质。

由于斜率对于所有的 a 来说都恒定为 -1 和 1，所以对于两个绝对值函数 $f_1,f_2(a_1\le a_2)$ 来说：

若 $x<\dfrac{a_1+a_2}{2}$，则 $f_1(x)>f_2(x)$，否则 $f_1(x)<f_2(x)$。

应用到 kruskal 的过程中，如果一条边 e_1 在 q_i 等于某个值时替代了以往 MST 中的 e_2 这条边，那么不论 e_1 在 E_L 内还是 E_R 内，e_2 都不会再被加入到 MST 中。

「JOISC 2022 Day4」复兴计划

考虑绝对值函数 $f(x)=|x-a|$ 的性质。

由于斜率对于所有的 a 来说都恒定为 -1 和 1，所以对于两个绝对值函数 $f_1,f_2(a_1\le a_2)$ 来说：

若 $x<\dfrac{a_1+a_2}{2}$，则 $f_1(x)>f_2(x)$，否则 $f_1(x)<f_2(x)$。

应用到 kruskal 的过程中，如果一条边 e_1 在 q_i 等于某个值时替代了以往 MST 中的 e_2 这条边，那么不论 e_1 在 E_L 内还是 E_R 内，e_2 都不会再被加入到 MST 中。

所以我们得到了边存在的性质：对于一条边 e_i 来说，在 q_i 增大到某一阈值时，e_i 通过替代某一条边被加入到 MST 中，在此之前都不在其内。而当 q_i 继续增大到另一阈值时，e_i 被另一条被替代，在此之后都不参与 MST。

也就是说，使得 e_i 在其对应的 MST 中的 q_i 构成一个区间 $[l_i,r_i]$。

「JOISC 2022 Day4」复兴计划

如果对于每个 e_i，求出了 $[l_i, r_i]$，那么计算答案是简单的。
按边权从小到大加入边，维护当前的 MST。

「JOISC 2022 Day4」复兴计划

如果对于每个 e_i，求出了 $[l_i, r_i]$，那么计算答案是简单的。

按边权从小到大加入边，维护当前的 MST。

当加入边 $e_i(u_i \xrightarrow{val_i} v_i)$ 时，若 u_i 与 v_i 不联通，则将其加入 MST。

否则，我们需要在当前树上 $u_i \rightsquigarrow v_i$ 的路径上选择一条边 e_j 将其替代。

「JOISC 2022 Day4」复兴计划

如果对于每个 e_i，求出了 $[l_i, r_i]$，那么计算答案是简单的。

按边权从小到大加入边，维护当前的 MST。

当加入边 $e_i(u_i \xrightarrow{val_i} v_i)$ 时，若 u_i 与 v_i 不联通，则将其加入 MST。

否则，我们需要在当前树上 $u_i \rightsquigarrow v_i$ 的路径上选择一条边 e_j 将其替代。

为了保证生成树边权最小，替代的边肯定是路径上边权最小的边，因为权越小，绝对值造成的贡献越大。

那么 e_i 替代 e_j 的时刻就是 $r_j = l_i = \dfrac{val_i + val_j}{2}$。

「JOISC 2022 Day4」复兴计划

如果对于每个 e_i，求出了 $[l_i, r_i]$，那么计算答案是简单的。

按边权从小到大加入边，维护当前的 MST。

当加入边 $e_i(u_i \xrightarrow{val_i} v_i)$ 时，若 u_i 与 v_i 不联通，则将其加入 MST。

否则，我们需要在当前树上 $u_i \rightsquigarrow v_i$ 的路径上选择一条边 e_j 将其替代。

为了保证生成树边权最小，替代的边肯定是路径上边权最小的边，因为权越小，绝对值造成的贡献越大。

那么 e_i 替代 e_j 的时刻就是 $r_j = l_i = \dfrac{val_i + val_j}{2}$。

至此我们可以贪心地求解出 $[l_i, r_i]$。由于需要支持路径最值，加边，断边，可以使用 LCT 维护。

复杂度 $O(m\log(n+m)+q)$。

「JOISC 2022 Day4」复兴计划

如果对于每个 e_i，求出了 $[l_i, r_i]$，那么计算答案是简单的。

按边权从小到大加入边，维护当前的 MST。

当加入边 $e_i(u_i \xrightarrow{val_i} v_i)$ 时，若 u_i 与 v_i 不联通，则将其加入 MST。

否则，我们需要在当前树上 $u_i \rightsquigarrow v_i$ 的路径上选择一条边 e_j 将其替代。

为了保证生成树边权最小，替代的边肯定是路径上边权最小的边，因为权越小，绝对值造成的贡献越大。

那么 e_i 替代 e_j 的时刻就是 $r_j = l_i = \dfrac{val_i + val_j}{2}$。

至此我们可以贪心地求解出 $[l_i, r_i]$。由于需要支持路径最值，加边，断边，可以使用 LCT 维护。

复杂度 $O(m\log(n+m)+q)$。

n 很小，所以此题暴力 Link-Cut 即可通过。

「JOISC 2023 Day3」合唱

题意简述：

给定长度为 $2n$ 的 01 序列，其中 n 个 0，n 个 1。

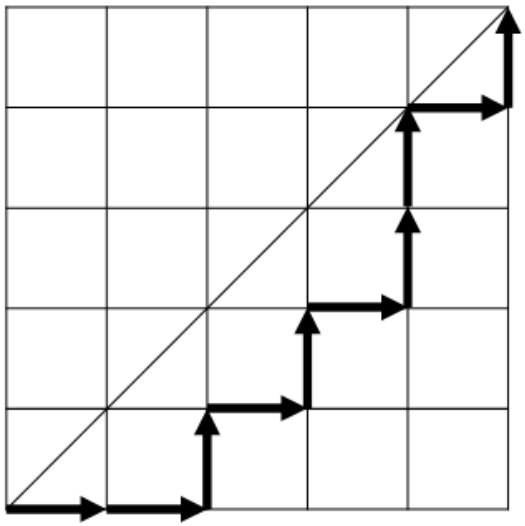
给定 m，问在至少多少次相邻项交换后，序列可以被恰好划分为 m 个子序列（非连续），每个子序列满足性质：

1. 子序列内 0 数量和 1 数量相同；
2. 所有的 1 在所有的 0 右边；
3. 即形态为：$00\cdots0011\cdots11$。

$n \leq 10^6$，但 $6s$。

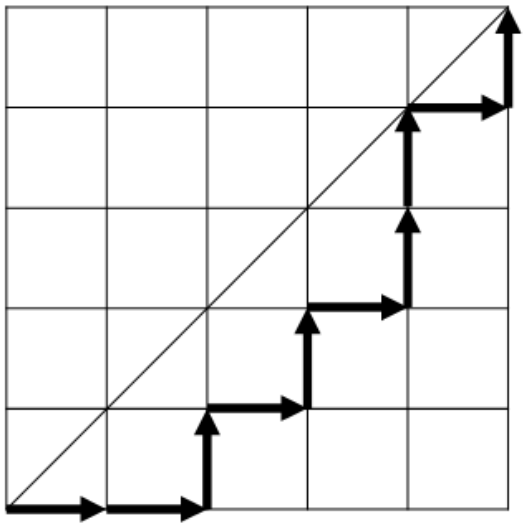
「JOISC 2023 Day3」合唱

考虑类似卡塔兰数地将其刻画为二维网格上的折线，0 代表向右走，1 代表向上走。



「JOISC 2023 Day3」合唱

考虑类似卡塔兰数地将其刻画为二维网格上的折线，0 代表向右走，1 代表向上走。



首先容易发现若存在答案，一定有 01 构成合法括号序列，即不超过对角线。

所以若原串不是合法括号序列，我们可以贪心 swap 将其强制合法。

「JOISC 2023 Day3」合唱

不考虑 m 的限制，有一个贪心构造解的方法：

每次选择任意数量（x）个最靠前的没被选择过的 0，选择最靠前的没被选择过的 1 与其匹配。

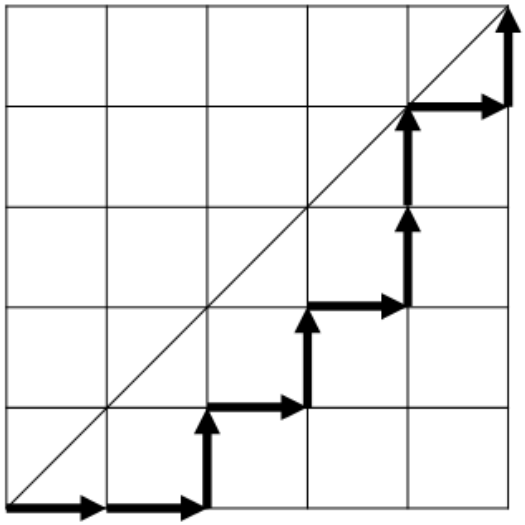
「JOISC 2023 Day3」合唱

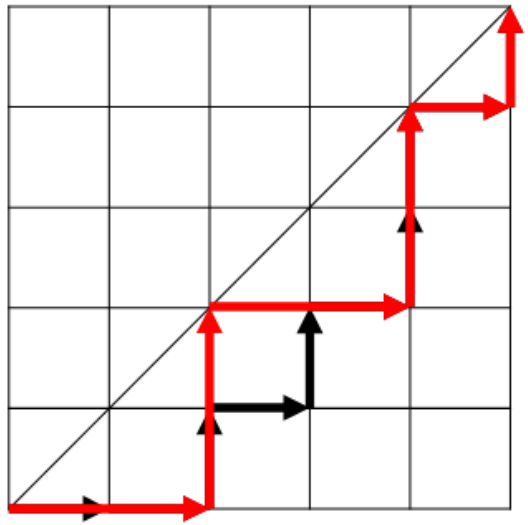
不考虑 m 的限制，有一个贪心构造解的方法：

每次选择任意数量（x）个最靠前的没被选择过的 0，选择最靠前的没被选择过的 1 与其匹配。

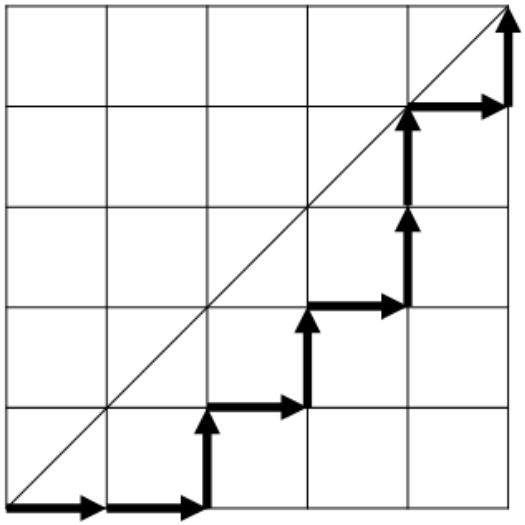
该方法合法，当且仅当每次选择的 0 的数量小于等于当前位置前缀 0 的数量。

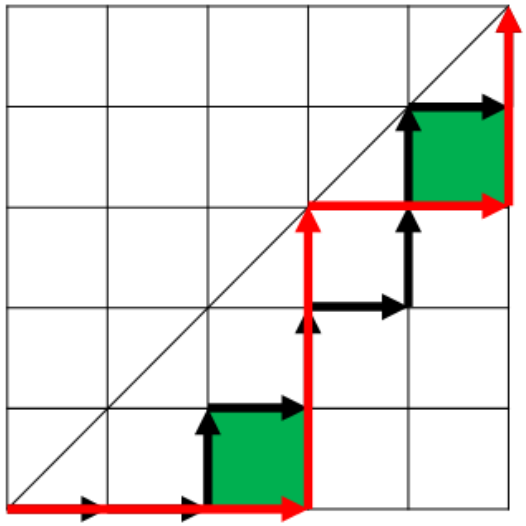
在图中体现为：若把每个子序列拎出来按顺序排成新序列（形态为 $000011110011\cdots$），图中新折线被原折线与对角线完全包含。





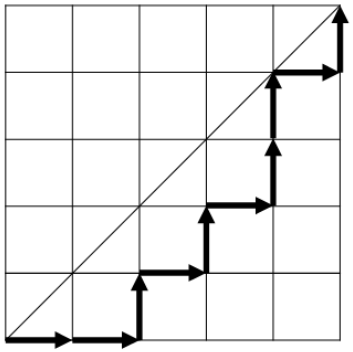
接下来加入修改，假设我们有一中并没有被完全包含的构造方案，我们可以通过 swap 操作使其恰好被包含：





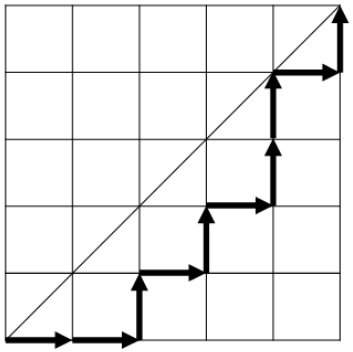
不难发现修改的代价即为超出原折线的面积。

「JOISC 2023 Day3」合唱



所以现在可以设计一个 DP，每次加入一个子序列，计算其贡献。

把网格的 $n+1$ 条竖线编号为 $0 \sim n$，定义 a_i 表示 i 这条竖线与原折线相交的最大高度（$0 \sim n$）。



所以现在可以设计一个 DP，每次加入一个子序列，计算其贡献。

把网格的 $n+1$ 条竖线编号为 $0 \sim n$，定义 a_i 表示 i 这条竖线与原折线相交的最大高度 $(0 \sim n)$。

记 $dp_{i,j}$ 表示现在子序列放到了 i 号竖线，放了 j 个子序列的最小代价。转移即为：

$$dp_{i,j} \leftarrow dp_{k,j-1} + \sum_{p=k}^{i-1} \max\{0, a_i - k\}$$

$$dp_{i,j} \leftarrow dp_{k,j-1} + \sum_{p=k}^{i-1} \max\{0, a_i - k\}$$

容易发现对于每个 k 来说，存在一个 $R(k)$ 满足 $a_{R(k)} - k \leq 0$（不存在即为 $k-1$）。

则令 pre_i 表示 a_i 的前缀和，转移系数可以优化到 $O(1)$ 计算：

$$dp_{i,j} \leftarrow dp_{k,j-1} + \begin{cases} 0 & R(k) \geq i-1 \\ pre_{i-1} - pre_{R(k)} - (i - R(k) - 1) \times k & \text{otherwise} \end{cases}$$

其中 $R(k)$ 由于是单调的，容易线性计算。

接下来，由于 dp 状态中出现了每次只增加 1 的第二维，可以证明其具有凸性。使用 wqs 二分即可优化。

接下来，由于 dp 状态中出现了每次只增加 1 的第二维，可以证明其具有凸性。使用 wqs 二分即可优化。

对于转移系数，可以证明其满足四边形不等式，故可以使用决策单调性对其优化。

复杂度为 $O(n\log^2 n)$，常数小可以过。

「JOISC 2023 Day3」合唱

接下来，由于 dp 状态中出现了每次只增加 1 的第二维，可以证明其具有凸性。使用 wqs 二分即可优化。

对于转移系数，可以证明其满足四边形不等式，故可以使用决策单调性对其优化。

复杂度为 $O(n\log^2 n)$，常数小可以过。

更加直观的做法，观察到转移式内有乘法，并且前缀和数组单调，故使用单调队列斜率优化即可解决。转移时进行判负取零即可

复杂度为 $O(n\log n)$。