

问题 A. 6789

输入文件：标准输入
输出文件：标准输出
时间限制：1 秒 1 秒 内存限制 1024 兆字节

Jaehyun 喜欢数字。在 10 个数字中，6、7、8 和 9 是他的最爱。因此，他专门制作了一套只有 6、7、8 和 9 的卡片。

目前，Jaehyun 有 $N \times M$ 张卡片。Jaehyun 想制作一个 N 乘 M 的神奇卡片矩阵。矩阵的每一行都要包含 M 张牌。他已经把纸牌摆成了 N 乘 M 矩阵的形状。

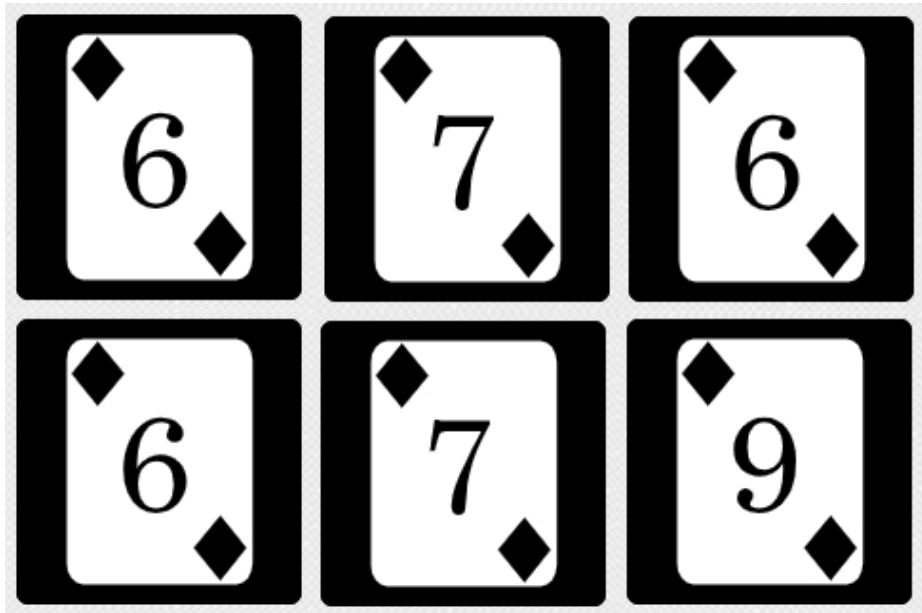


图 1. 初始状态，非点对称。

要成为魔法矩阵，矩阵必须是点对称的：将矩阵旋转 180 度，结果与原始矩阵相同。例如，8 与它本身是点对称的，而 6 和 9 彼此是点对称的。Jaehyun 不想转换纸牌的位置，所以他的目标是只旋转纸牌的原始位置，使矩阵点对称。

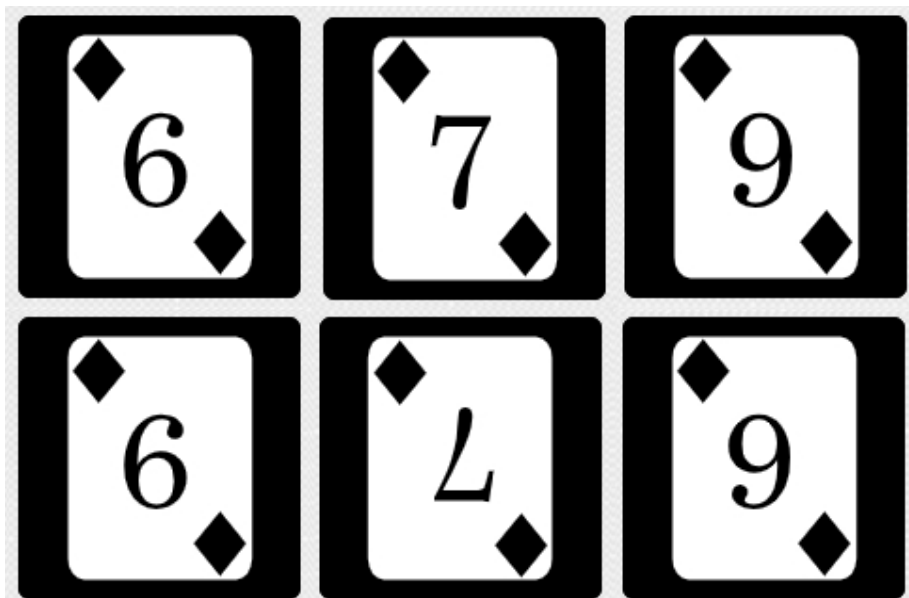


图 2. 旋转两张牌后，它们是点对称的。

找出最少要转动多少张牌才能组成神奇矩阵。

输入

第一行包含两个整数 N 和 M 。 ($1 \leq N, M \leq 500$)

接下来的 N 行中，每一行都包含一个由 M 个字符组成的字符串，表示写在每张卡片上的数字。保证每个字符都是 "6"、"7"、"8 "或 "9 "中的一个。

输出

在第一行打印制作魔术矩阵所需的最少翻牌张数。如果无法组成魔术矩阵，则打印 "-1"（不带引号）。

示例

标准输入	标准输出
2 3 676 679	2
3 3 888 888 888	0
1 1 7	-1

问题 B. 更大的推箱子游戏 40k

输入文件： *标准输入*
输出文件 *标准输出*
时间限制： 1 秒 1 秒 内存限制 1024
兆字节

推箱子是一款著名的益智游戏，玩家在 $N \times M$ 大小的网格中移动，并将 1×1 大小的箱子推到 1×1 大小的存储位置。

大推箱子是推箱子游戏的一种可能的变体，但箱子的大小和存储位置都比 1×1 大。这道题特别使用了 2×2 。

大型推箱子游戏的规则与推箱子游戏相同。网格中的每个方格都是一个空方格或一堵墙。一些面积为 2×2 的空方格中各有 2×2 大小的盒子，一些面积为 2×2 的空方格中各有 2×2 大小的存储位置。

玩家位于网格中，可以上下左右移动到相邻的空方格，但不能穿过墙壁、方框或网格外。如果玩家试图移动到一个方框内，该方框会被推到该方向的相邻方格内。箱子不能被推到其他箱子、墙壁或网格外，也不能被拉动。盒子的数量等于存储位置的数量。当所有的盒子都到达储存位置时，谜题就解开了。

您的任务是制作一个更大的推箱子方格，至少需要 **40,000** 次移动才能解开谜题。为了方便起见，网格必须满足以下限制条件：

- $1 \leq n, m, n + m \leq 100$ 。
- 网格包含一个盒子和一个存储位置。
- 播放器、盒子和存储位置不得相交。

输入

此问题没有输入。

输出

在第一行，打印两个空格分隔的整数 N 、 M ；它们描述了网格的大小。

在随后的 N 行中，每行打印一个长度为 M 的字符串；它描述网格的每一行。每个字符串必须由 $.$ 、 $\#$ 、 P 、 B 、 S 组成；每个字符分别表示空方格、墙壁、玩家、盒子、存储位置。

网格中必须包含一个 P 、四个 B 和四个 S 。 B 和 S 必须分别组成一个 2×2 的正方形。当然，网格必须是可解的。

请注意，示例输出只是为了演示格式良好的输出。由于它可以在少于 40,000 步内求解，因此并不是正确答案。

示例

标准输入	标准输出
	5 6SSSS .#BB#. .BB.P

问题 C. 清洁

输入文件： *标准输入*
输出文件： *标准输出*
时间限制 2 秒 内存限制 1024
兆字节

Minje 经营着一家游戏厅。其中一款游戏位于大小为 $N \times M$ 的网格中。每个单元格中都写有 L、R、U、D 四个字符中的一个。这分别表示左/右/上/下方向。玩家可以在网格内自由移动，但**不能**朝自己所在单元格所写的方向移动，也不能移动到网格外。

Minje 会在玩家进入和离开网格后清理网格。然而，清理所有 $N \times M$ 个单元格是很累的。因此，Minje 只想清理玩家可能会访问的单元格。

有 Q 名玩家参与了游戏。Minje 会记住每个玩家在游戏中站过的第一个格子和最后一个格子。这些单元格不一定位于角落或边缘。请计算 Minje 应该清理的单元格数量。每局游戏都是独立的，不会影响其他游戏的结果。

输入

在第一行中，给出了三个整数 N 、 M 、 Q ，分别表示网格的大小和玩家的数量。($1 \leq N, M \leq 1\,000, 1 \leq Q \leq 300\,000$)

在接下来的 N 行中，给出了大小为 M 的字符串，表示写在每个网格单元格中的字符。每个字符都是这四个字母中的一个：L、R、U、D。

在接下来的 Q 行中，给出了四个整数 $x_1, y_1, x_2, y_{(2)}$ 。这表示玩家在单元格 $(x_1, y_{(1)})$ 中开始游戏，在单元格 $(x_2, y_{(2)})$ 中结束游戏。($1 \leq x_{(1)}, x_2 \leq N, 1 \leq y_{(1)}, y_2 \leq M$)

输出

在每行 Q 中，按输入的顺序打印表示查询答案的单个整数。查询的答案是

- 如果无法从单元格 (x_1, y_1) 移动到单元格 (x_2, y_2) ，则打印 0。
- 否则，打印 Minje 应该清理的单元格数量。

示例

标准输入	标准输出
5 5 5	0
DDDDD	14
RDDDL	20
RRDLL	14
RUUUL	5
UUUUU	
1 1 5 5	
2 2 5 5	
3 3 5 5	
4 4 5 5	
5 5 5 5	

问题 D. 容器

输入文件：标准输入
输出文件：标准输出
时间限制：2.5 秒 内存限制
1024 兆字节

Cki86201 Container Creator (CCC) 生产两种集装箱。一种容量为 1 吨，另一种容量为 2 吨。CCC 目前在一个仓库里有 N 个集装箱，它们排成一排。在装运集装箱时，最好把先装运的集装箱放在前面。因此，CCC 打算重新排列这 N 个集装箱。

为了重新排列集装箱，我们使用了机器。机器可以选择两个或三个连续的集装箱，并颠倒它们的顺序。使用机器的成本是所选容器的总容量加上基本成本 C 。因此，颠倒两个容器 $[2, 1]$ 需要成本 $3 + C$ ，颠倒三个容器 $[1, 1, 2]$ 需要成本 $4 + C$ 。

虽然这项任务对 cki86201 来说微不足道，但他现在必须玩《守望先锋》。请找出以最小成本重新排列集装箱的方法。

输入

第一行给出了两个整数 N, C 。 ($1 \leq N \leq 500, 0 \leq C \leq 1000$)

下一行是大小为 N 的字符串，由两个字符 "1"、"2" 组成。这表示仓库中按**当前**排列顺序存放的集装箱的容量。

下一行是一个大小为 N 的字符串，由两个字符 "1"、"2" 组成。这表示仓库中按**预期**排列顺序存放的集装箱的容量。

保证两个字符串中 "1" 和 "2" 的数量相等。

输出

在第一行，打印数字 K ，表示您使用机器的次数。

在接下来的 K 行中，打印两个整数 i, j ($1 \leq i < j \leq N, j - i \leq 2$)，表示您已使用机器将 $i, i+1, \dots, j$ -容器的当前状态。

如果答案是有效的（即用输出的阶次进行模拟，会产生所需的排列），并且使用了尽可能小的成本，那么答案将被视为正确。

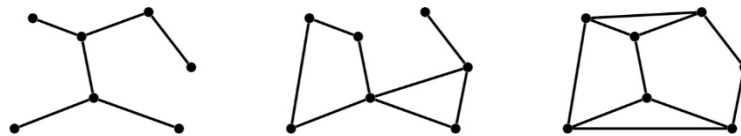
示例

标准输入	标准输出
5 2 11221 21112	2 1 3 4 5
7 0 2212121 1212122	4 6 7 4 6 2 4 1 2
7 2 2212121 1212122	3 1 3 3 5 5 7

问题 E.死亡仙人掌协会

输入文件：标准输入
输出文件：标准输出
时间限制：10 秒 10 秒 内存限制
1024 兆字节

在韩国，*宠物图*已经成为替代动物宠物的热门选择。其中，*树木*因其温和的特性和可爱的外形，成为最受欢迎的宠物图形选择。然而，Koo 在这个行业中却以特立独行而闻名，因为他只为顾客提供*仙人掌图*。*树状图*和*仙人掌图*是某一类图。树图是没有循环的连通图。*仙人掌图*是一种连通图，其中不存在属于两个不同循环的边。让我们来看一个例子。

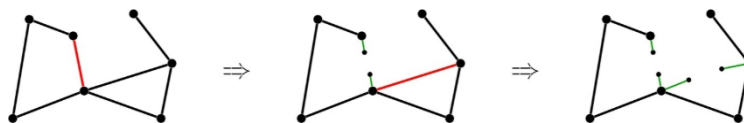


第一幅图中的图形既是树也是仙人掌，第二幅图中的图形是仙人掌，最后一幅图中的图形既不是树也不是仙人掌。更多循环，更多问题。

Koo 带着热情和爱心为仙人掌图形服务了 21 年，但现在他老了，也累了。虽然宠物图的流行让他店里的客人越来越多，但他们很少成为顾客，因为大多数人认为仙人掌图对宠物来说很难看。辜鸿铭的财运越来越差，生活也越来越艰难。最后，Koo 接受了残酷的现实，他决定开始做树图生意--把他可爱的仙人掌剪掉边缘。

图形是一种非常敏感的生物，砍掉中间的边缘会使整个图形衰败。但是，如果你精确地切断顶点和边的交汇点，图形就能自我修复。因此，你应该去掉整条边，而不是切掉中间。此外，切勿将图形切成断开状态：这对图形来说是非常可怕的。

如果切掉边 $e = \{u, v\}$ ，顶点 u, v 就会受伤。仙人掌是一种顽强的生物，因此它能立即愈合受伤的伤疤，长出新的边和新的末端顶点。每一条边和每一个顶点都有一个非负的**愈合因子**，顾氏事先知道这个**因子**。如果一条边 e 的愈合因子为 RE_e ，而一个受伤的顶点 v 的愈合因子为 RV_v ，那么每一个受伤的顶点都会创建一条长度为 $RE_e + RV_v$ 的新边，同时在边的另一侧创建一个新顶点。



在图中，红色的边是我们剪切的边，绿色的边是新创建的边。

如果我们从仙人掌的每个循环中切下一条边，我们就可以创造出一棵树。一般来说，人们喜欢直径小的树，树的直径是树上两个顶点之间最短路径的最大可能长度。为了赚大钱，Koo 想砍掉仙人掌，使其直径最小。给定一个仙人掌，请帮助 Koo 找出切割循环中的边所能得到的最小直径。

输入

第一行给出了两个整数 N, M ，分别表示仙人掌的顶点和边的数量。($3 \leq n \leq 100\,000$, $n \leq m \leq 150\,000$)。每个顶点都标有数字 $1, 2, \dots, N$ ，每条边分别标有 $1, 2, \dots, M$ 。

下一行给出 N 个整数 RV_1, RV_2, \dots, RV_N 表示每个顶点的愈合因子。 ($0 \leq RV(i) \leq 10^9$).

在接下来的 M 行中，第 i 行给出了四个整数 $A_{(i)}, B_i, L_{(i)}, RE_{(i)}$ 。 这表示第 i 条边连接两个顶点 $A_{(i)}, B_{(i)}$ 且边长为 $L_{(i)}$ ，并具有愈合因子 $RE_{(i)}$ 。 ($1 \leq A_{(i)}, B_i \leq N, A_i \neq B_{(i)}, 1 \leq L_{(i)}, RE_{(i)} \leq 10^9$)

保证给定的图是仙人掌图，且没有连接同一对顶点的不同边。

输出

打印一个表示答案的整数。

示例

标准输入	标准输出
3 3 1 2 3 3 1 2 3 1 2 1 2 2 3 3 1	10
5 6 1 2 3 4 5 1 2 6 1 1 3 5 4 2 3 4 2 1 4 3 6 1 5 2 3 4 5 1 5	22

问题 F. 希尔伯特旅馆

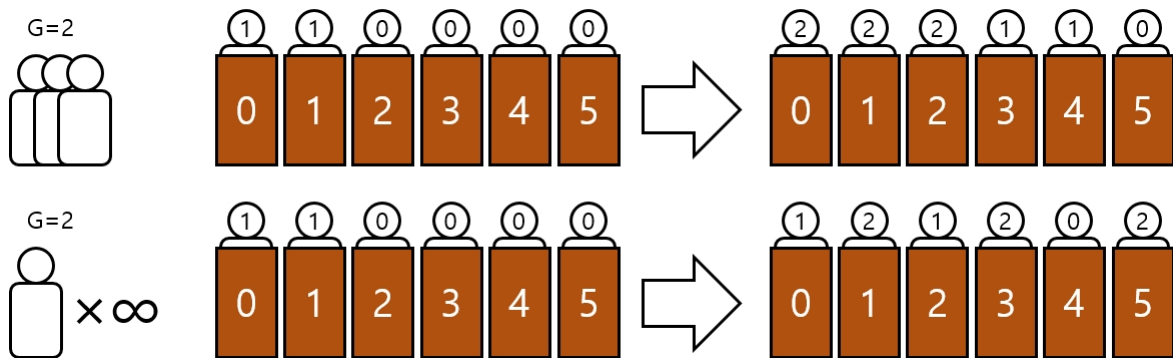
输入文件：	标准输入
输出文件	标准输出
时间限制	1.5 秒 内存限制
	1024 兆字节

希尔伯特的旅馆有无数个房间，分别编号为 0、1、2……。每个房间最多住一位客人。由于人们往往成群结队地办理入住手续，所以酒店有一个成群结队的柜台变量 G 。

希尔伯特的酒店今天盛大开业。希尔伯特的酒店今天隆重开业。不久，无穷多的人同时到来，挤满了酒店的每个房间。所有客人的分组编号都是 0， G 被设为 1。

具有讽刺意味的是，即使每个房间都住满了人，酒店也能接纳更多的客人：

- 如果有 k ($k \geq 1$) 人来到酒店，那么对于每个房间号 x ， x 号房间的客人都会搬到 x 号房间 + k 。之后，新客人会住满从 0 到 $k-1$ 的所有房间。
- 如果有无限多的人入住酒店，那么对于每个房间号 x ， x 号房间的客人都会搬到 $2x$ 号房间。之后，新客人会住满所有奇数房间。



你必须编写一个程序来处理下列查询：

- 1 k - 如果 $k \geq 1$ ，则有 k 人到达酒店。如果 $k = 0$ ，则无穷多的人到达酒店。给新客人分配组号 G ，然后将 G 递增 1。
- 2 g x - 找出包含组号为 g 的客人的第 x 个最小房间号，并将其输出到 modulo $10^9 + 7$ ，然后换行。
- 3 x - 输出 x 号房间客人的组号，换行。

输入

第一行给出一个整数 Q ($1 \leq Q \leq 300,000$)，表示查询次数。接下来的每一行都包含一个查询。查询中的所有数字都是整数。

- 对于 1 k 个查询， $0 \leq k \leq 10^9$ 。
- 对于 2 g x 查询， $g < G$ ， $1 \leq x \leq 10^9$ ，至少有 x 位客人的组号为 g 。
- 对于 3 x 查询， $0 \leq x \leq 10^9$ 。

输出

处理所有查询并按要求输出。保证输出不为空。

示例

标准输入	标准输出
10	0
3 0	1
1 3	0
2 1 2	9
1 0	4
3 10	4
2 2 5	
1 5	
1 0	
3 5	
2 3 3	

注释

如果你知道 "红心"，请假设 "无限 "仅指 "可数无限"。如果不知道，就不必担心。

问题 G. 词法最小行走

输入文件： 标准输入
输出文件： 标准输出
时间限制：2 秒 2 秒 内存限制 1024
兆字节

有一个有向图 G ，它有 N 个节点和 M 条边。每个节点编号为 1 到 N ，每条边编号为 1 到 M 。对于每个 i ($1 \leq i \leq M$)，边 i 从顶点 u_i 到顶点 v_i ，并具有**唯一**的颜色 $c_{(i)}$ 。

行走的定义是边 e_1, e_2, \dots, e_l 序列，其中每 $1 \leq k < l$ ， $v_{(e_k)}$ (边 e_k 的尾) 与 $u_{(e_{k+1})}$ (边 e_{k+1} 的头) 相同。我们可以说一条行走 e_1, e_2, \dots, e_l 开始于顶点 $u_{(e_1)}$ 并结束于顶点 $v_{(e_l)}$ 。请注意，同一条边可以在一次行走中出现多次。

行走 e_1, e_2, \dots, e_l 的**颜色序列**定义为 $c_{(e_1)}, c_{(e_2)}, \dots, c_{(e_l)}$ 。

考虑 G 中从顶点 S 到顶点 T 的所有长度不超过 10^{100} 的彩色行走序列。

输入

输入的第一行包含四个空格分隔的整数 N, M, S 和 T ($1 \leq N \leq 100\,000, 0 \leq M \leq 300\,000, 1 \leq S \leq N, 1 \leq T \leq N, S \neq T$)。

接下来是 M 条线：其中第 j ($1 \leq j \leq M$) 条线包含三个空间隔离整数 u_i, v_i 和 $c_{(i)}$ ($1 \leq u_i, v_i \leq N, u_i \neq v_i, 1 \leq c_{(i)} \leq 10^9$)；它描述了一条从顶点 u_i 到顶点 v_i 的带颜色 $c_{(i)}$ 的定向边。

图中没有多条边，每条边都有唯一的颜色。形式上，对于任意 $1 \leq i < j \leq M$ ， $c_i \neq c_j$ 和 $(u_i, v_i) \neq (u_j, v_j)$ 成立。

输出

如果不存在从顶点 S 到顶点 T 的行走，则打印 "IMPOSSIBLE"。(不带引号)

否则，假设 a_1, a_2, \dots, a_l 是所有从顶点 S 到顶点 T 长度至多为 10^{100} 的颜色序列中的词典最小序列。

- 如果 $l \leq 10^6$ ，则在第一行打印 a_1, a_2, \dots, a_l 。每个打印的整数之间应有一个空格。
- 如果 $l > 10^6$ ，打印 "TOO LONG" (太长)。(不带引号)

示例

标准输入	标准输出
3 3 1 3 1 2 1 2 3 7 1 3 5	1 7
3 4 1 3 1 2 1 2 1 2 2 3 7 1 3 5	过长
2 0 2 1	不可能

备注

序列 p_1, p_2, \dots, p_n 在词法上小于另一序列 q_1, q_2, \dots, q_m ，当且仅当下列条件之一成立时

成立：

- 存在唯一的 j ($0 \leq j < \min(n, m)$)，其中 $p_1 = q_1, p_2 = q_2, \dots, p_j = q_j$ 和 $p_{(j)+1} < q_{(j)+1}$ 。
- $n < m$ 和 $p_1 = q_1, p_2 = q_2, \dots, p_n = q_n$ 。换句话说， p 是 q 的严格前缀。

问题 H. 最大化

输入文件： 标准输入
输出文件 标准输出
时间限制 2 秒 内存限制 1024
兆字节

Maximizer 有两个排列组合 $A = [a_1, a_2, \dots, a_N]$, $B = [b_1, b_2, \dots, b_N]$ 。 A 和 B 的长度均为 N 并由 1 到 N 之间的不同整数组成。

最大化者希望最大化每个元素的差值之和 $\sum_{i=1}^N |a_i - b_i|$ 。但他只能确切地说，他只能交换 A 中相邻两个元素的 a_i 和 $a_{(i)+1}$ ，交换对象为 1 到 $N-1$ 中的某个 i 。他想交换多少次就交换多少次。

要使差和最大，最少需要多少次交换？

输入

第一行包含一个整数 N 。 ($1 \leq N \leq 250000$)

第二行包含 N 个整数 $a_1, a_2, \dots, a_{(N)}$ ($1 \leq a_i \leq N$)。第三行包含 N 个整数 $b_1, b_2, \dots, b_{(N)}$ ($1 \leq b_i \leq N$)。

每一个 $[a_1, a_2, \dots, a_N]$ 和 $[b_1, b_2, \dots, b_N]$ 都是一个排列。换句话说，它由从 1 到 N 的不同整数组成。

输出

打印一个整数，即最大化差值和所需的最小交换次数。

示例

标准输入	标准输出
3 1 2 3 1 2 3	2
4 3 4 1 2 3 2 4 1	3

问题 I. 最小直径生成树

输入文件：标准输入
输出文件：标准输出
时间限制 5 秒 内存限制 1024
兆字节

给您一个简单相连的无向加权图 G ，其中有 N 个节点和 M 条边。每个节点编号为 1 到 N 。

G 的生成树是 G 的一个子图，它是一棵树，连接着 G 的所有顶点。 G 的最小直径生成树是具有最小直径的 G 的生成树。

编写一个程序，找出任何直径最小的生成树。

输入

输入的第一行包含两个整数 N ($2 \leq N \leq 500$) 和 M ($N-1 \leq M \leq \frac{N(N-1)}{2}$)。那么下面是 M 行：第 i ($1 \leq i \leq M$)-行包含三个空格分隔的整数 u_i , v_i 和 $l_{(i)}$

($1 \leq u_i, v_i \leq N$, $1 \leq l_i \leq 10^9$); 它描述了连接顶点 u_i 和顶点 v_i 的长度为 $l_{(i)}$ 的双向边。

保证给定的图没有任何循环或多条边，并且图是连通的。

输出

第一行，打印 G 的最小直径生成树的直径。

在接下来的 $N-1$ 行中，打印 G 的最小直径生成树中的边的描述。第 j ($1 \leq j \leq N-1$)-行应包含两个空格分隔的整数 x_i 和 y_i ($1 \leq x_i, y_i \leq N$)；它描述了连接顶点 x_i 和 y_i 的双向边。

如果有多个可能的答案，请打印其中任意一个。

示例

标准输入	标准输出
3 3 1 2 1 2 3 1 3 1 1	2 1 2 3 1
6 7 1 2 10 2 3 20 1 3 30 3 4 1000 4 5 30 5 6 20 4 6 10	1060 3 4 6 4 5 6 2 3 1 2

问题 J. 百草园

输入文件：标准输入
输出文件：标准输出
时间限制：1.5 秒 内存限制
1024 兆字节



图 1. 阴天中的加普川和世博桥

加普川是流经大德创新城的一条小溪：它是大田的一个研究区，包括韩国科学技术院、世博科学公园、国立科学博物馆等。加普川的水边被用作公园，是休闲和娱乐设施。

在这个问题中，我们将 *Gapcheon* 建模为一条略微弯曲的弧线。在弧线上，每厘米正好有 10^6 个点。在 *Gapcheon* 中，有 N 座桥将弧线上的两个不同点用直线段连接起来。这样的线段可能在端点处与其他线段相交，但绝不会与其他线段相交。对于每一对点，最多存在一座直接连接这两点的桥。

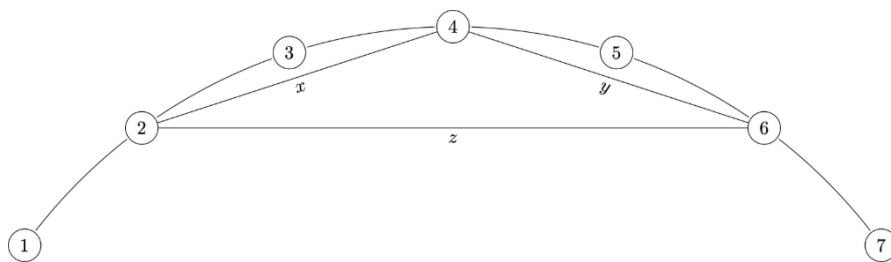


图 2: x 、 y 、 z 是不相交的桥，只在端点处相互接触。这可能是一个输入实例。编号为 $8 \dots 10^{(6)}$ 的点为简洁起见省略。

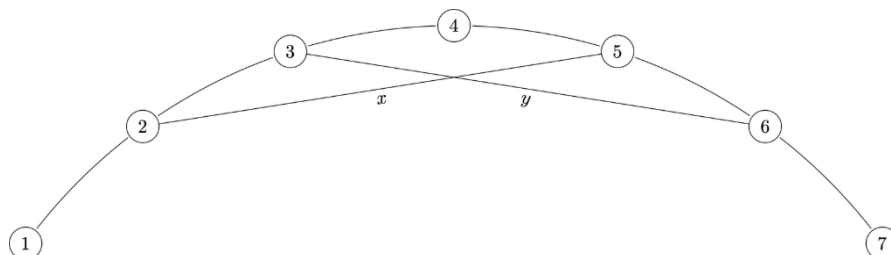


图 3. x 、 y 是相互交叉的桥梁。这不是一个可能的输入实例。编号为 $8 \dots 10^{(6)}$ 的点为简洁起见省略。

市议会计划在这些桥梁上安装一些照明灯，使加普川成为一个在夜晚更令人愉悦的地方。市议会为每座桥梁计算了安装灯光后的美学价值。这些值可以用正整数表示。

然而，过多的灯光会在午夜时分打扰居民。为了解决这个问题，议会决定制定一些规定：对于相邻两点之间的每条弧线，从那里最多只能看到 k 座亮灯的桥梁。当线段的一个端点的索引最多为 i ，而线段的另一个端点的索引至少为 $i+1$ 时，我们称从连接 i ($i+1$) 的弧线上可以看到该线段。

市议会希望考虑光污染和夜景之间的权衡，因此您应提供所有整数 $1 \leq k \leq N$ 的最大可能的美学价值总和。

输入

第一行包含一个整数 N 。（ $1 \leq N \leq 250\,000$ ）

接下来的 N 行包含三个整数 $S_i, E_i, V_{(i)}$ ，表示有一条直线桥连接点 $S_{(i)}, E_{(i)}$ ，并具有美学值 $V_{(i)}$ 。（ $1 \leq S_i < E_i \leq 10^6, 1 \leq V_i \leq 10^9$ ）。

保证没有线段连接同一点，也没有两条不同的线段交叉。

输出

打印用空格分隔的 N 个整数。如果 $k=i$ ，则第 i 个整数（ $1 \leq i \leq N$ ）应为答案。

示例

标准输入	标准输出
<pre> 6 1 2 10 2 3 10 1 3 21 3 4 10 4 5 10 3 5 19 </pre>	<pre> 41 80 80 80 80 80 </pre>
<pre> 4 1 5 1 2 5 1 3 5 1 4 5 1 </pre>	<pre> 1 2 3 4 </pre>

注

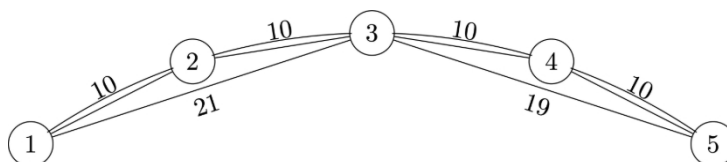


图 4. 输入样本 1 的描述。

问题 K. 变化之风

输入文件：	标准输入
输出文件：	标准输出
时间限制	12 秒 内存限制
	1024 兆字节

这个问题的原标题是 "无一点的树积公制 Voronoi 图查询"。

给你两棵大小为 N 的加权树 T_1, T_2 ，其中每个顶点都标有数字 $1..N$ 。设 $dist(T_1, i, j)$ 是树 T_1 中从节点 i 到 j 的最短路径的总权重，类似地定义 $dist(T_2, i, j)$ 。

考虑大小为 N 的点集。与曼哈顿度量相似（事实上，这是曼哈顿度量的概括），我们可以定义两点之间的距离 $1 \leq i, j \leq N$ ：它是两个距离之和，即 $dist(T_1, i, j) + dist(T_2, i, j)$ 。对于每个 $1 \leq i \leq N$ ，请确定离点 "最近的点"。
 i 。形式上，对于每个 i ，您应该找到 $\min_{(j)} dist(T_1, i, j) + dist(T_2, i, j)$ 。

输入

第一行，给出一个整数 N ，表示两棵树的顶点数。 ($2 \leq N \leq 250\,000$)

在接下来的 $N-1$ 行中，将给出第一棵树的描述。 $N-1$ 行中的每一行都包含三个整数 S_i, E_i, W_i ，表示有一条边连接两个顶点 S_i, E_i 且权重为 W_i ($1 \leq S_i, E_i \leq N, 1 \leq W_i \leq 10^9$)

在接下来的 $N-1$ 行中，将以相同格式对第二棵树进行描述。

输出

打印 N 行，每行包含一个整数。在第 i 行，应打印一个整数，表示第 i 点的答案。

示例

标准输入	标准输出
5 1 2 10 2 4 20 3 4 30 4 5 50 1 2 15 1 3 25 1 4 35 1 5 25	25 25 85 65 105
9 5 7 6577 4 5 8869 5 9 9088 2 1 124 6 2 410 2 8 8154 4 8 4810 3 4 4268 3 9 763 6 2 8959 7 4 7984 3 8 504 8 6 9085 5 2 4861 1 9 8539 1 7 7834	18084 9369 9582 23430 26694 9369 23430 9582 22988