

Problem A. 最优乘车

Time limit 1000 ms

Mem limit 131072 kB

Description

H 城是一个旅游胜地，每年都有成千上万的人前来观光。为方便游客，巴士公司在各个旅游景点及宾馆，饭店等地都设置了巴士站并开通了一些单程巴士线路。每条单程巴士线路从某个巴士站出发，依次途经若干个巴士站，最终到达终点巴士站。

一名旅客最近到 H 城旅游，他很想去看 S 公园游玩，但如果从他所在的饭店没有一路巴士可以直接到达 S 公园，则他可能要先乘某一路巴士坐几站，再下来换乘同一站台的另一路巴士，这样换乘几次后到达 S 公园。

现在用整数 $1, 2, \dots, N$ 给 H 城的所有的巴士站编号，约定这名旅客所在饭店的巴士站编号为 1，S 公园巴士站的编号为 N 。

写一个程序，帮助这名旅客寻找一个最优乘车方案，使他在从饭店乘车到 S 公园的过程中换车的次数最少。

Input

第一行有两个数字 M 和 N ($1 \leq M \leq 100, 1 < N \leq 500$)，表示开通了 M 条单程巴士线路，总共有 N 个车站。

从第二行到第 M 行依次给出了第 1 条到第 M 条巴士线路的信息。其中第 $i + 1$ 行给出的是第 i 条巴士线路的信息，从左至右按运行顺序依次给出了该线路上的所有站号相邻两个站号之间用一个空格隔开。

Output

只有一行。如果无法乘巴士从饭店到达 S 公园，则输出 **NO**；

否则输出你的程序所找到的最少换车次数，换车次数为 0 表示不需换车即可到达。

Sample 1

Input	Output
3 7 6 7 4 7 3 6 2 1 3 5	2

Problem B. 香甜的黄油 Sweet Butter

Time limit 1000 ms

Mem limit 131072 kB

Description

Farmer John 发现了做出全威斯康辛州最甜的黄油的方法：糖。

把糖放在一片牧场上，他知道 N 只奶牛会过来舔它，这样就能做出能卖好价钱的超甜黄油。当然，他将付出额外的费用在奶牛上。

Farmer John 很狡猾。像以前的 Pavlov，他知道他可以训练这些奶牛，让它们在听到铃声时去一个特定的牧场。他打算将糖放在那里然后下午发出铃声，以至他可以在晚上挤奶。

Farmer John 知道每只奶牛都在各自喜欢的牧场（一个牧场不一定只有一头牛）。给出各头牛在的牧场和牧场间的路线，找出使所有牛到达的路程和最短的牧场（他将把糖放在那）。

Input

第一行包含三个整数 N, P, C ，分别表示奶牛数、牧场数和牧场间道路数。

第二行到第 $N + 1$ 行，每行一个整数，其中第 i 行的整数表示第 $i - 1$ 头奶牛所在的牧场号。

第 $N + 2$ 行到第 $N + C + 1$ 行，每行包含三个整数 A, B, D ，表示牧场号为 A 和 B 的两个牧场之间有一条长度为 D 的双向道路相连。

Output

输出一行一个整数，表示奶牛必须行走的最小的距离和。

Sample 1

Input	Output
3 4 5 2 3 4 1 2 1 1 3 5 2 3 7 2 4 3 3 4 5	8

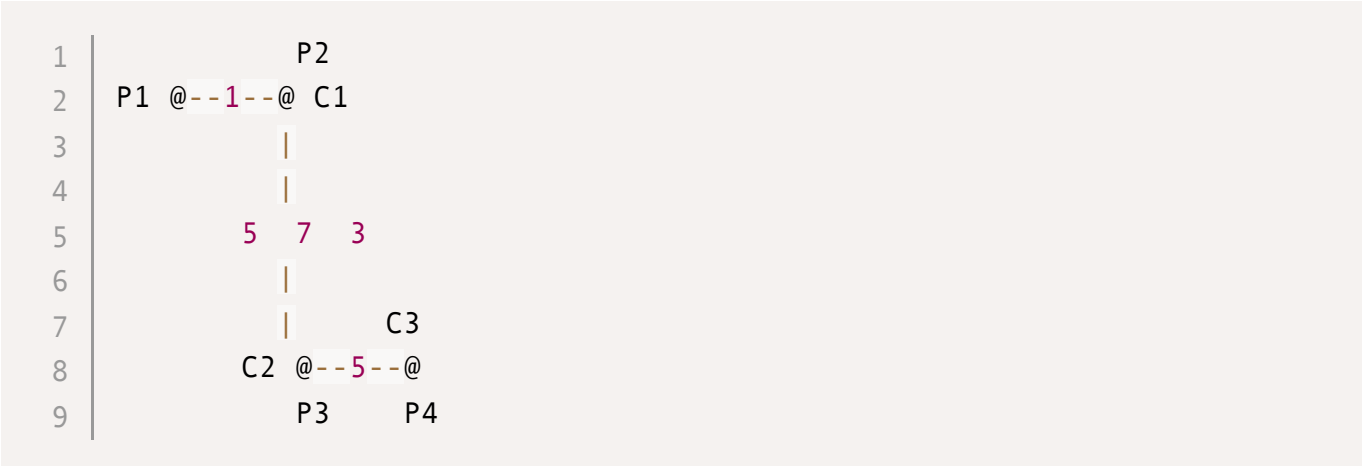
Hint

数据范围

对于所有数据， $1 \leq N \leq 500$ ， $2 \leq P \leq 800$ ， $1 \leq A, B \leq P$ ， $1 \leq C \leq 1450$ ， $1 \leq D \leq 255$ 。

样例解释

作图如下：



把糖放在4号牧场最优。

Problem C. Missile Silos

Time limit 2000 ms

Mem limit 262144 kB

Input file `stdin`

Output file `stdout`

名为Berland的国家由编号从1到 n 的整数号城市组成。其中一些城市通过双向道路相连。每条道路都有一定的长度。通过这些道路，任意两个城市之间都存在路径。根据一些超级机密文件，Berland由超级机密导弹保护。超级机密导弹的确切位置保密，但Bob设法获取了这些信息。该信息表明，所有导弹发射井距离首都的距离恰好为 l 。首都位于编号为 s 的城市。

文件给出了正式定义：当且仅当某个位置（可以是城市或道路上的某一点）到首都的最短距离恰好等于 l 时，超级机密导弹发射井位于该位置。Bob想知道Berland中有多少个导弹发射井，以便将这些信息出售给敌方间谍。帮帮Bob。

输入

第一行包含三个整数 n 、 m 和 s ($2 \leq n \leq 10^5$, $n - 1 \leq m \leq \min(10^5, \frac{n(n-1)}{2})$, $1 \leq s \leq n$)，分别表示城市数量、国家中的道路数量和首都的编号。首都位于第 s 号城市。

接下来的 m 行包含道路的描述。每条道路由三个整数 v_i 、 u_i 、 w_i ($1 \leq v_i, u_i \leq n$, $v_i \neq u_i$, $1 \leq w_i \leq 1000$) 描述，其中 v_i 、 u_i 是该道路连接的城市编号， w_i 是道路的长度。最后一行包含整数 l ($0 \leq l \leq 10^9$)，表示首都到导弹发射井的距离。保证：

- 任意两个城市之间最多只有一条道路；
- 每条道路连接两个不同的城市；
- 从每个城市到其他任何城市都至少有一条道路可通行。

输出

输出一个数字，表示位于Berland的超级机密导弹发射井的数量。

示例

Input	Output
4 6 1 1 2 1 1 3 3 2 3 1 2 4 1 3 4 1 1 4 2 2	3

Input	Output
5 6 3 3 1 1 3 2 1 3 4 1 3 5 1 1 2 6 4 5 8 4	3

注意

在第一个示例中，导弹发射井位于城市3和4以及道路(1, 3)上，距离城市1分别为2（对应地，距离城市3为1）。

在第二个示例中，一个导弹发射井位于道路(1, 2)的正中间。另外两个发射井位于道路(4, 5)上，距离城市4向城市5方向分别为3，距离城市5向城市4方向分别为3。

Problem D. Time Travel

Time limit 2000 ms

Mem limit 524288 kB

伯兰是一个拥有悠久历史的国家，几个世纪以来一直在修建和毁坏道路。已知伯兰一直有 n 个城市。你还记录了国家历史上从1到 t 编号的 t 个关键时刻。每个记录包含一份在特定时刻可用于在伯兰旅行的城市对之间的双向道路清单。

你发现了一台可以将你传送到关键时刻的时间机器。不幸的是，你无法选择最终会到达的时间点，但你知道由 k 个时刻组成的顺序 a_i ，时间机器将会传送你。由于旅行之间时间很短，当你发现自己在下一个关键时刻（**包括最后一次时间旅行之后**）时，你最多可以沿着那个时刻的一条现有道路旅行，从之前时间旅行前所在的城市出发。

当前，你在城市1，时间机器已经将你传送到时刻 a_1 。你想尽快到达城市 n 。确定到达城市 n 所需的最小时间旅行次数，**包括第一次**。

输入

第一行包含两个整数 n 和 t ($2 \leq n \leq 2 \cdot 10^5, 1 \leq t \leq 2 \cdot 10^5$) — 伯兰城市的数量和关于国家历史关键时刻的记录数量。接下来是每个第 t 个记录的描述。

每个记录描述的第一行包含一个整数 m_i ($0 \leq m_i \leq \min\left(\frac{n(n-1)}{2}, 2 \cdot 10^5\right)$) — 第 i 个记录中道路数量。

接下来的 m_i 行中，每行包含两个整数 v_j 和 u_j ($1 \leq v_j, u_j \leq n, v_j \neq u_j$) — 第 i 个关键历史时刻中第 j 条道路连接的城市编号。

下一行包含一个整数 k ($1 \leq k \leq 2 \cdot 10^5$) — 移动之间的时间时刻数量。

下一行包含 k 个整数 $a_1, a_2, \dots, a_k$ ($1 \leq a_i \leq t$) — 每次移动后你所在的时间时刻。

保证 m_i 的总和不超过 $2 \cdot 10^5$ 。保证每个无序对 (u, v) 在道路描述中最多出现一次。

输出

输出一个整数 — 到达城市 n 从城市1所需的最小时间旅行次数，如果不可能则输出 -1 。

注意到移动到时间时刻 a_1 也被视为一次移动。

示例1

Input	Output
5 2 4 1 2 2 3 3 4 4 5 2 2 3 3 5 6 2 1 2 1 2 1	5

示例2

Input	Output
5 2 3 1 2 3 1 4 3 2 2 1 4 5 5 1 2 1 1 1	-1

注意

在第一个示例中，你在城市1，移动到时刻 $a_1 = 2$ 。由于没有适合通过的道路，你什么也不做，移动到时刻 $a_2 = 1$ ，之后你沿着道路(1, 2)旅行。移动到时刻 $a_3 = 2$ ，你沿着道路(2, 3)旅行。移动到时刻 $a_4 = 1$ ，你留在城市3并进行下一次时间旅行。在时刻 $a_5 = 2$ ，你沿着道路(3, 5)旅行，并在经过5次时间旅行后到达最终城市。

在第二个示例中，可以证明使用给定的时间旅行无法到达城市5。

Problem E. Paths and Trees

Time limit 3000 ms

Mem limit 262144 kB

小女孩苏茜意外地发现了她哥哥的笔记本。她有很多事情要做，比解决问题更重要，但她觉得这个问题太有趣了，所以她想知道它的解决方案，并决定向你询问。因此，问题陈述如下。

假设我们有一个**连通的**加权无向图 $G = (V, E)$ （这里 V 是顶点集， E 是边集）。从顶点 u 到其他顶点的最短路径树是这样的图 $G_1 = (V, E_1)$ ，它是初始图 E 的边集 E_1 的子集，并且从 u 到任何顶点和到 G 、 G_1 的最短路径长度相同。

给定一个**连通的**加权无向图 G 和顶点 u 。你的任务是找到给定图的从顶点 u 出发的最短路径树，其边的总权重是可能的最小值。

输入

第一行包含两个数字， n 和 m ($1 \leq n \leq 3 \cdot 10^5, 0 \leq m \leq 3 \cdot 10^5$) —— 图的顶点数和边数。

接下来的 m 行，每行包含三个整数，表示一条边 —— u_i, v_i, w_i —— 连接的顶点编号和边的权重 ($u_i \neq v_i, 1 \leq w_i \leq 10^9$)。保证图是连通的，且**任意一对顶点之间最多只有一条边**。

输入的最后一行包含整数 u ($1 \leq u \leq n$) —— 起始顶点的编号。

输出

第一行输出树的边的最小总权重。

接下来一行，按顺序打印包含在树中的边的索引，用空格分隔。边的编号从1开始。你可以以任意顺序打印边的编号。

如果存在多个答案，可以输出其中任意一个。

示例

Input	Output
3 3 1 2 1 2 3 1 1 3 2 3	2 1 2

Input	Output
4 4 1 2 1 2 3 1 3 4 1 4 1 2 4	4 2 3 4

注意

在第一个示例中，存在两种可能的最短路径树：

- 包含边1 – 3和2 – 3（总权重为3）；
- 包含边1 – 2和2 – 3（总权重为2）；

例如，包含边1 – 2和1 – 3的树不会是顶点3的最短路径树，因为在这棵树中，从顶点3到顶点2的距离等于3，而在原始图中是1。

Problem F. Jzzhu and Cities

Time limit 2000 ms

Mem limit 262144 kB

Input file `stdin`

Output file `stdout`

Jzzhu是A国的总统。他的国家有 n 个城市，编号从1到 n 。城市1是A国的首都。此外，有 m 条道路连接这些城市。可以通过第 i 条道路从城市 u_i 到 v_i （反之亦然），这条道路的长度为 x_i 。最后，这个国家有 k 条火车路线。可以使用第 i 条火车路线从国家首都到城市 s_i （反之亦然），这条路线的长度为 y_i 。

Jzzhu不想浪费国家的钱，所以他打算关闭一些火车路线。请告诉Jzzhu在以下条件下可以关闭的最大火车路线数量：从每个城市到首都的最短路径长度不能改变。

输入

第一行包含三个整数 n, m, k ($2 \leq n \leq 10^5$; $1 \leq m \leq 3 \cdot 10^5$; $1 \leq k \leq 10^5$)。

接下来的 m 行，每行包含三个整数 u_i, v_i, x_i ($1 \leq u_i, v_i \leq n$; $u_i \neq v_i$; $1 \leq x_i \leq 10^9$)。

接下来的 k 行，每行包含两个整数 s_i 和 y_i ($2 \leq s_i \leq n$; $1 \leq y_i \leq 10^9$)。

保证每个城市到首都至少有一条路径。注意，两个城市之间可能有多条道路。此外，可能有多条路线从首都到同一个城市。

输出

输出一个整数，表示可以关闭的最大火车路线数量。

示例1

Input	Output
5 5 3 1 2 1 2 3 2 1 3 3 3 4 4 1 5 5 3 5 4 5 5 5	2

示例2

Input	Output
2 2 3 1 2 2 2 1 3 2 1 2 2 2 3	2

Problem G. Counting Shortcuts

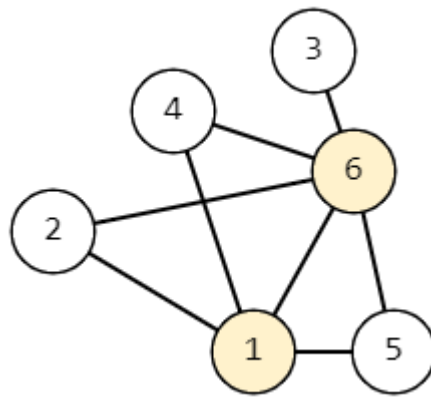
Time limit 2000 ms

Mem limit 262144 kB

背景

给定一个无向连通图，包含 n 个顶点和 m 条边。该图不包含自环（从一个顶点到自身的边）和多重边（即每对顶点之间不超过一条边）。图的顶点编号从 1 到 n 。

找到从顶点 s 到 t 的路径数量，这些路径的长度与从 s 到 t 的最短路径长度之差不超过 1。需要考虑所有合适的路径，即使它们经过同一个顶点或边多次（即它们不是简单路径）。



由 6 个顶点和 8 条边组成的图

例如，设 $n = 6$ 、 $m = 8$ 、 $s = 6$ 和 $t = 1$ ，并让图的形状如上图所示。那么从 s 到 t 的最短路径长度为 1。考虑所有长度最多为 $1 + 1 = 2$ 的路径。

- $6 \rightarrow 1$ 。路径的长度为 1。
- $6 \rightarrow 4 \rightarrow 1$ 。路径长度为 2。
- $6 \rightarrow 2 \rightarrow 1$ 。路径长度为 2。
- $6 \rightarrow 5 \rightarrow 1$ 。路径长度为 2。

总共有 4 条符合条件的路径。

输入

测试的第一行包含数字 t ($1 \leq t \leq 10^4$) — 测试用例的数量。

在每个测试用例之前，有一行空行。

测试用例的第一行包含两个数字 n, m ($2 \leq n \leq 2 \cdot 10^5, 1 \leq m \leq 2 \cdot 10^5$) — 图中顶点和边的数量。

第二行包含两个数字 s 和 t ($1 \leq s, t \leq n, s \neq t$) — 路径的起始和结束顶点的编号。

接下来的 m 行包含边的描述：第 i 行包含两个整数 u_i, v_i ($1 \leq u_i, v_i \leq n$) — 连接第 i 条边的顶点编号。可以保证图是连通的，并且不包含自环和多重边。

可以保证所有测试用例输入数据中值 n 的总和不超过 $2 \cdot 10^5$ 。同样，可以保证所有测试用例输入数据中值 m 的总和不超过 $2 \cdot 10^5$ 。

输出

对于每个测试用例，输出一个数字 — 从 s 到 t 的路径数量，使得它们的长度与最短路径的长度之差不超过 1。

由于这个数字可能太大，输出时取模 $10^9 + 7$ 。

示例

Input	Output
4	2
4 4	4
1 4	1
1 2	11
3 4	
2 3	
2 4	
6 8	
6 1	
1 4	
1 6	
1 5	
1 2	
5 6	
4 6	
6 3	
2 6	
5 6	
1 3	
3 5	
5 4	
3 1	
4 2	
2 1	
1 4	
8 18	
5 1	
2 1	
3 1	
4 2	
5 2	
6 5	
7 3	
8 4	
6 4	
8 7	
1 4	
4 7	
1 6	
6 7	
3 8	
8 5	
4 5	
4 3	
8 2	