

题目描述

展开

春春是一名道路工程师，负责铺设一条长度为 n 的道路。

铺设道路的主要工作是填平下陷的地表。整段道路可以看作是 n 块首尾相连的区域，一开始，第 i 块区域下陷的深度为 d_i 。

春春每天可以选择一段连续区间 $[L, R]$ ，填充这段区间中的每块区域，让其下陷深度减少 1。在选择区间时，需要保证，区间内的每块区域在填充前下陷深度均不为 0。

春春希望你能帮他设计一种方案，可以在最短的时间内将整段道路的下陷深度都变为 0。

输入格式

输入文件包含两行，第一行包含一个整数 n ，表示道路的长度。第二行包含 n 个整数，相邻两数间用一个空格隔开，第 i 个整数为 d_i 。

输出格式

输出文件仅包含一个整数，即最少需要多少天才能完成任务。

输入输出样例

输入 #1

复制

6
4 3 2 5 3 5

输出 #1

复制

9

说明/提示

- 【样例解释】
- 一种可行的最佳方案是，依次选择： $[1, 6]$ 、 $[1, 6]$ 、 $[1, 2]$ 、 $[1, 1]$ 、 $[4, 6]$ 、 $[4, 4]$ 、 $[4, 4]$ 、 $[6, 6]$ 、 $[6, 6]$ 。
- 【数据规模与约定】
- 对于 30% 的数据， $1 \leq n \leq 10$ ；
对于 70% 的数据， $1 \leq n \leq 1000$ ；
对于 100% 的数据， $1 \leq n \leq 100000, 0 \leq d_i \leq 10000$ 。

题目描述

[展开](#)

在网友的国度中共有 n 种不同面额的货币，第 i 种货币的面额为 $a[i]$ ，你可以假设每一种货币都有无穷多张。为了方便，我们把货币种数为 n 、面额数组为 $a[1..n]$ 的货币系统记作 (n, a) 。

在一个完善的货币系统中，每一个非负整数的金额 x 都应该可以被表示出，即对每一个非负整数 x ，都存在 n 个非负整数 $t[i]$ 满足 $a[i] \times t[i]$ 的和为 x 。然而，在网友的国度中，货币系统可能是不完善的，即可能存在金额 x 不能被该货币系统表示出。例如在货币系统 $n = 3, a = [2, 5, 9]$ 中，金额 $1, 3$ 就无法被表示出来。

两个货币系统 (n, a) 和 (m, b) 是等价的，当且仅当对于任意非负整数 x ，它要么均可以被两个货币系统表出，要么不能被其中任何一个表出。

现在网友们打算简化一下货币系统。他们希望找到一个货币系统 (m, b) ，满足 (m, b) 与原来的货币系统 (n, a) 等价，且 m 尽可能的小。他们希望你来协助完成这个艰巨的任务：找到最小的 m 。

输入格式

输入文件的第一行包含一个整数 T ，表示数据的组数。

接下来按照如下格式分别给出 T 组数据。每组数据的第一行包含一个正整数 n 。接下来一行包含 n 个由空格隔开的正整数 $a[i]$ 。

输出格式

输出文件共有 T 行，对于每组数据，输出一行一个正整数，表示所有与 (n, a) 等价的货币系统 (m, b) 中，最小的 m 。

输入输出样例

输入 #1

[复制](#)

```
2
4
3 19 10 6
5
11 29 13 19 17
```

输出 #1

[复制](#)

```
2
5
```

说明/提示

在第一组数据中，货币系统 $(2, [3, 10])$ 和给出的货币系统 (n, a) 等价，并可以验证不存在 $m < 2$ 的等价的货币系统，因此答案为 2。在第二组数据中，可以验证不存在 $m < n$ 的等价的货币系统，因此答案为 5。

【数据范围与约定】

测试点	n	a_i	测试点	n	a_i
1	$= 2$	≤ 1000	11	≤ 13	≤ 16
2			12		
3			13		
4	$= 3$		14	≤ 25	≤ 40
5			15		
6			16		
7	$= 4$		17	≤ 100	≤ 25000
8			18		
9	$= 5$		19		
10			20		

对于 100% 的数据，满足 $1 \leq T \leq 20, n, a[i] \geq 1$ 。

题目描述

[展开](#)

C 城将要举办一系列的赛车比赛。在比赛前，需要在城内修建 m 条赛道。

C 城一共有 n 个路口，这些路口编号为 $1, 2, \dots, n$ ，有 $n - 1$ 条适合于修建赛道的双向通行的道路，每条道路连接着两个路口。其中，第 i 条道路连接的两个路口编号为 a_i 和 b_i ，该道路的长度为 l_i 。借助这 $n - 1$ 条道路，从任何一个路口出发都能到达其他所有的路口。

一条赛道是一组互不相同的道路 e_1, e_2, \dots, e_k ，满足可以从某个路口出发，依次经过道路 e_1, e_2, \dots, e_k （每条道路经过一次，不允许调头）到达另一个路口。一条赛道的长度等于经过的各道路的长度之和。为保证安全，要求每条道路至多被一条赛道经过。

目前赛道修建的方案尚未确定。你的任务是设计一种赛道修建的方案，使得修建的 m 条赛道中长度最小的赛道长度最大（即 m 条赛道中最短赛道的长度尽可能大）

输入格式

输入文件第一行包含两个由空格分隔的正整数 n, m ，分别表示路口数及需要修建的赛道数。

接下来 $n - 1$ 行，第 i 行包含三个正整数 a_i, b_i, l_i ，表示第 i 条适合于修建赛道的道路连接的两个路口编号及道路长度。保证任意两个路口均可通过这 $n - 1$ 条道路相互到达。每行中相邻两数之间均由一个空格分隔。

输出格式

输出共一行，包含一个整数，表示长度最小的赛道长度的最大值。

输入输出样例

输入 #1

复制

7 1
1 2 10
1 3 5
2 4 9
2 5 8
3 6 6
3 7 7

输出 #1

复制

31

输入 #2

复制

9 3
1 2 6
2 3 3
3 4 5
4 5 10
6 2 4
7 2 9
8 4 7
9 4 4

输出 #2

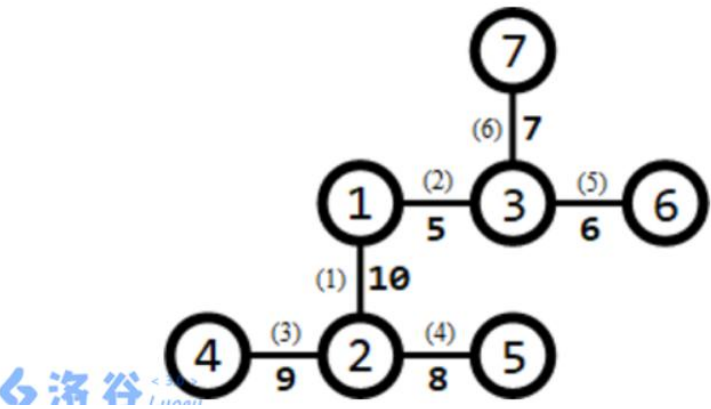
复制

15

说明/提示

【输入输出样例 1 说明】

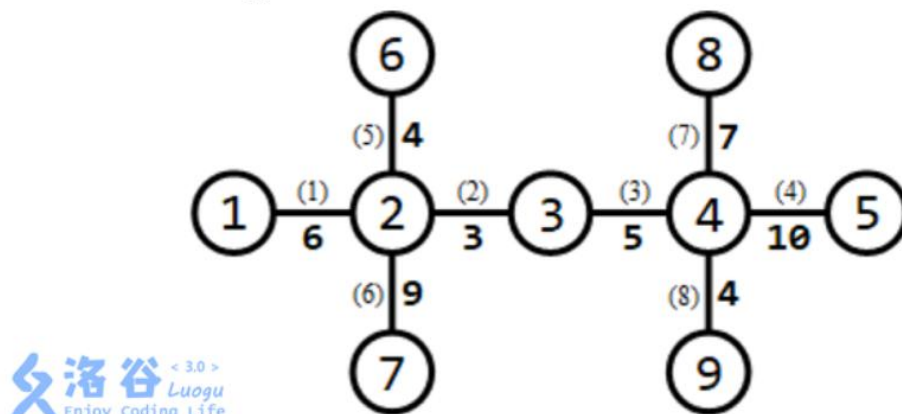
所有路口及适合于修建赛道的道路如下图所示：



道路旁括号内的数字表示道路的编号，非括号内的数字表示道路长度。需要修建 1 条赛道。可以修建经过第 3, 1, 2, 6 条道路的道路（从路口 4 到路口 7），则该赛道的长度为 $9 + 10 + 5 + 7 = 31$ ，为所有方案中的最大值。

【输入输出样例 2 说明】

所有路口及适合于修建赛道的道路如下图所示：



需要修建 3 条赛道。可以修建如下 3 条赛道：

1. 经过第 1, 6 条道路的道路（从路口 1 到路口 7），长度为 $6 + 9 = 15$ ；
2. 经过第 5, 2, 3, 8 条道路的道路（从路口 6 到路口 9），长度为 $4 + 3 + 5 + 4 = 16$ ；
3. 经过第 7, 4 条道路的道路（从路口 8 到路口 5），长度为 $7 + 10 = 17$ 。长度最小的赛道长度为 15，为所有方案中的最大值。

数据规模与约定

所有测试数据的范围和特点如下表所示：

测试点编号	n	m	$a_i = 1$	$b_i = a_i + 1$	分支不超过 3
1	≤ 5	$= 1$	否	否	是
2	≤ 10	$\leq n - 1$	否	是	是
3	≤ 15	$\leq n - 1$	是	否	否
4	$\leq 10^3$	$= 1$	否	否	是
5	$\leq 3 \times 10^4$	$= 1$	是	否	否
6	$\leq 3 \times 10^4$	$= 1$	否	否	否
7	$\leq 3 \times 10^4$	$\leq n - 1$	是	否	否
8	$\leq 5 \times 10^4$	$\leq n - 1$	是	否	否
9	$\leq 10^3$	$\leq n - 1$	否	是	是
10	$\leq 3 \times 10^4$	$\leq n - 1$	否	是	是
11	$\leq 5 \times 10^4$	$\leq n - 1$	否	是	是
12	≤ 50	$\leq n - 1$	否	否	是
13	≤ 50	$\leq n - 1$	否	否	是
14	≤ 200	$\leq n - 1$	否	否	是
15	≤ 200	$\leq n - 1$	否	否	是

16	$\leq 10^3$	$\leq n - 1$	否	否	是
17	$\leq 10^3$	$\leq n - 1$	否	否	否
18	$\leq 3 \times 10^4$	$\leq n - 1$	否	否	否
19	$\leq 3 \times 10^4$	$\leq n - 1$	否	否	否
20	$\leq 5 \times 10^4$	$\leq n - 1$	否	否	否

其中，「分支不超过 3」的含义为：每个路口至多有 3 条道路与其相连。

对于所有的数据， $2 \leq n \leq 5 \times 10^4$ ， $1 \leq m \leq n-1$ ， $1 \leq a_i, b_i \leq n$ ， $1 \leq l_i \leq 10^4$ 。

题目描述

 展开

小 Y 是一个爱好旅行的 OIer。她来到 X 国，打算将各个城市都玩一遍。

小 Y 了解到，X 国的 n 个城市之间有 m 条双向道路。每条双向道路连接两个城市。不存在两条连接同一对城市的道路，也不存在一条连接一个城市和他本身的道路。并且，从任意一个城市出发，通过这些道路都可以到达任意一个其他城市。小 Y 只能通过这些道路从一个城市前往另一个城市。

小 Y 的旅行方案是这样的：任意选定一个城市作为起点，然后从起点开始，每次可以选择一条与当前城市相连的道路，走向一个没有去过的城市，或者沿着第一次访问该城市时经过的道路后退到上一个城市。当小 Y 回到起点时，她可以选择结束这次旅行或继续旅行。需要注意的是，小 Y 要求在旅行方案中，每个城市都被访问到。

为了让自己的旅行更有意义，小 Y 决定在每到达一个新的城市（包括起点）时，将它的编号记录下来。她知道这样会形成一个长度为 n 的序列。她希望这个序列的字典序最小，你能帮帮她吗？对于两个长度均为 n 的序列 A 和 B ，当且仅当存在一个正整数 x ，满足以下条件时，我们说序列 A 的字典序小于 B 。

- 对于任意正整数 $1 \leq i < x$ ，序列 A 的第 i 个元素 A_i 和序列 B 的第 i 个元素 B_i 相同。
- 序列 A 的第 x 个元素的值小于序列 B 的第 x 个元素的值。

输入格式

输入文件共 $m + 1$ 行。第一行包含两个整数 $n, m (m \leq n)$ ，中间用一个空格分隔。

接下来 m 行，每行包含两个整数 $u, v (1 \leq u, v \leq n)$ ，表示编号为 u 和 v 的城市之间有一条道路，两个整数之间用一个空格分隔。

输出格式

输出格式

输出文件包含一行， n 个整数，表示字典序最小的序列。相邻两个整数之间用一个空格分隔。

输入输出样例

输入 #1

复制

```
6 5
1 3
2 3
2 5
3 4
4 6
```

输出 #1

复制

```
1 3 2 5 4 6
```

输入 #2

复制

```
6 6
1 3
2 3
2 5
3 4
4 5
4 6
```

输出 #2

复制

```
1 3 2 4 5 6
```

说明/提示

【数据规模与约定】

对于 100% 的数据和所有样例， $1 \leq n \leq 5000$ 且 $m = n-1$ 或 $m = n$ 。

对于不同的测试点，我们约定数据的规模如下：

对于不同的测试点，我们约定数据的规模如下：

测试点编号	$n =$	$m =$	特殊性质
1, 2, 3	10	$n - 1$	无
4, 5	100		无
6, 7, 8	1000		每个城市最多与两个城市相连
9, 10	1000		无
11, 12, 13	5000		每个城市最多与三个城市相连
14, 15	5000		无
16, 17	10	n	无
18, 19	100		无
20, 21, 22	1000		每个城市最多与两个城市相连
23, 24, 25	5000		无

题目描述

展开

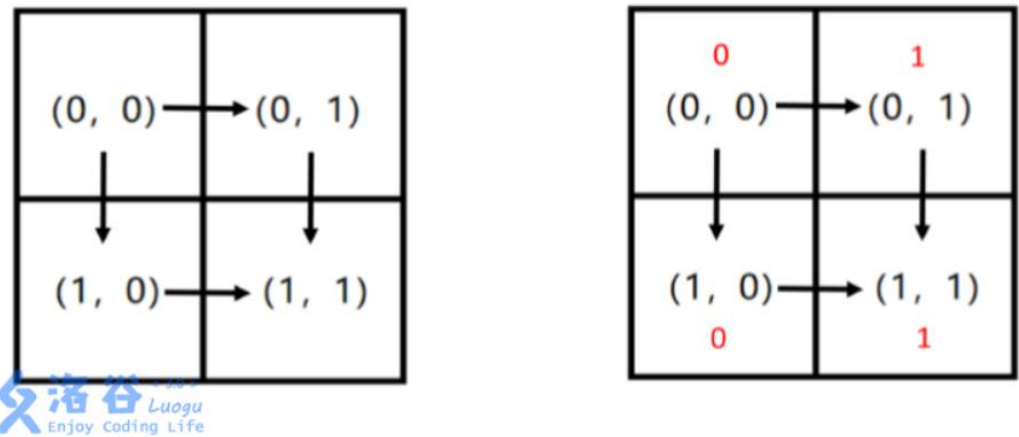
小 D 特别喜欢玩游戏。这一天，他在玩一款填数游戏。

这个填数游戏的棋盘是一个 $n \times m$ 的矩形表格。玩家需要在表格的每个格子中填入一个数字（数字 0 或者数字 1），填数时需要满足一些限制。

下面我们来具体描述这些限制。

为了方便描述，我们先给出一些定义：

- 我们用每个格子的行列坐标来表示一个格子，即（行坐标，列坐标）。（注意：行列坐标均从 0 开始编号）
- 合法路径 P ：一条路径是合法的当且仅当：
 - 这条路径从矩形表格的左上角的格子 $(0, 0)$ 出发，到矩形的右下角格子 $(n - 1, m - 1)$ 结束；
 - 在这条路径中，每次只能从当前的格子移动到右边与它相邻的格子，或者从当前格子移动到下面与它相邻的格子。例如：在下面这个矩形中，只有两条路径是合法的，它们分别是 $P_1: (0, 0) \rightarrow (0, 1) \rightarrow (1, 1)$ 和 $P_2: (0, 0) \rightarrow (1, 0) \rightarrow (1, 1)$ 。



对于一条合法的路径 P ，我们可以用一个字符串 $w(P)$ 来表示，该字符串的长度为 $n + m - 2$ ，其中只包含字符“R”或者字符“D”，第 i 个字符记录了路径 P 中第 i 步的移动方法，“R”表示移动到当前格子右边与它相邻的格子，“D”表示移动到当前格子下面与它相邻的格子。例如，上图中对于路径 P_1 ，有 $w(P_1) = "RD"$ ；而对于另一条路径 P_2 ，有 $w(P_2) = "DR"$ 。

同时，将每条合法路径 P 经过的每个格子上填入的数字依次连接后，会得到一个长度为 $n + m - 1$ 的 01 字符串，记为 $s(P)$ 。例如，如果我们在格子 $(0, 0)$ 和 $(1, 0)$ 上填入数字 0，在格子 $(0, 1)$ 和 $(1, 1)$ 上填入数字 1（见上图红色数字）。那么对于路径 P_1 ，我们可以得到 $s(P_1) = "011"$ ，对于路径 P_2 ，有 $s(P_2) = "001"$ 。

游戏要求小 D 找到一种填数字 0、1 的方法，使得对于两条路径 P_1, P_2 ，如果 $w(P_1) > w(P_2)$ ，那么必须 $s(P_1) \leq s(P_2)$ 。我们说字符串 a 比字符串 b 小，当且仅当字符串 a 的字典序小于字符串 b 的字典序，字典序的定义详见第一题。但是仅仅是找一种方法无法满足小 D 的好奇心，小 D 更想知道这个游戏有多少种玩法，也就是说，有多少种填数字的方法满足游戏的要求？

小 D 能力有限，希望你帮助他解决这个问题，即有多少种填 0、1 的方法能满足题目要求。由于答案可能很大，你需要输出答案对 $10^9 + 7$ 取模的结果。

输入格式

输入文件共一行，包含两个正整数 n, m ，由一个空格分隔，表示矩形的大小。其中 n 表示矩形表格的行数， m 表示矩形表格的列数。

输出格式

输出共一行，包含一个正整数，表示有多少种填 0、1 的方法能满足游戏的要求。注意：输出答案对 $10^9 + 7$ 取模的结果。

输入输出样例

输入 #1	复制	输出 #1	复制
2 2		12	
输入 #2	复制	输出 #2	复制
3 3		112	
输入 #3	复制	输出 #3	复制
5 5		7136	

说明/提示

【样例解释】

说明/提示

【样例解释】

洛谷 < 3.0 >
Luogu
Enjoy coding Life

<table><tr><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	0	0	0	0	<table><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td></tr></table>	0	0	1	0	<table><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	0	1	1	0	<table><tr><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table>	0	0	0	1
0	0																		
0	0																		
0	0																		
1	0																		
0	1																		
1	0																		
0	0																		
0	1																		
<table><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	0	0	1	1	<table><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table>	0	1	1	1	<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	1	0	0	0	<table><tr><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td></tr></table>	1	0	1	0
0	0																		
1	1																		
0	1																		
1	1																		
1	0																		
0	0																		
1	0																		
1	0																		
<table><tr><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	1	1	1	0	<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table>	1	0	0	1	<table><tr><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	1	0	1	1	<table><tr><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table>	1	1	1	1
1	1																		
1	0																		
1	0																		
0	1																		
1	0																		
1	1																		
1	1																		
1	1																		

【数据规模与约定】

测试点编号	$n \leq$	$m \leq$
1~4	3	3
5~10	2	1000000
11~13	3	1000000
14~16	8	8
17~20	8	1000000



题目描述

[展开](#)

Z 国有 n 座城市， $(n - 1)$ 条双向道路，每条双向道路连接两座城市，且任意两座城市都能通过若干条道路相互到达。

Z 国的国防部长小 Z 要在城市中驻扎军队。驻扎军队需要满足如下几个条件：

- 一座城市可以驻扎一支军队，也可以不驻扎军队。
- 由道路直接连接的两座城市中至少要有一座城市驻扎军队。
- 在城市里驻扎军队会产生花费，在编号为 i 的城市中驻扎军队的花费是 p_i 。

小 Z 很快就规划出了一种驻扎军队的方案，使总花费最小。但是国王又给小 Z 提出了 m 个要求，每个要求规定了其中两座城市是否驻扎军队。小 Z 需要针对每个要求逐一给出回答。具体而言，如果国王提出的第 j 个要求能够满足上述驻扎条件（不需要考虑第 j 个要求之外的其它要求），则需要给出在此要求前提下驻扎军队的最小开销。如果国王提出的第 j 个要求无法满足，则需要输出 -1 。现在请你来帮助小 Z。

输入格式

第一行有两个整数和一个字符串，依次表示城市数 n ，要求数 m 和数据类型 $type$ 。 $type$ 是一个由大写字母 A，B 或 C 和一个数字 1，2，3 组成的字符串。它可以帮助你获得部分分。你可能不需要用到这个参数。这个参数的含义在【数据规模与约定】中有具体的描述。

第二行有 n 个整数，第 i 个整数表示编号 i 的城市中驻扎军队的花费 p_i 。

接下来 $(n - 1)$ 行，每行两个整数 u, v ，表示有一条 u 到 v 的双向道路。

接下来 m 行，每行四个整数 a, x, b, y ，表示一个要求是在城市 a 驻扎 x 支军队，在城市 b 驻扎 y 支军队。其中， x, y 的取值只有 0 或 1：

- 若 x 为 0, 表示城市 a 不得驻扎军队。
- 若 x 为 1, 表示城市 a 必须驻扎军队。
- 若 y 为 0, 表示城市 b 不得驻扎军队。
- 若 y 为 1, 表示城市 b 必须驻扎军队。

输入文件中每一行相邻的两个数据之间均用一个空格分隔。

输出格式

输出共 m 行, 每行包含一个个整数, 第 j 行表示在满足国王第 j 个要求时的最小开销, 如果无法满足国王的第 j 个要求, 则该行输出 -1 。

输入输出样例

输入 #1

复制

```
5 3 C3
2 4 1 3 9
1 5
5 2
5 3
3 4
1 0 3 0
2 1 3 1
1 0 5 0
```

输出 #1

复制

```
12
7
-1
```

说明/提示

样例 1 解释

- 对于第一个要求, 在 4 号和 5 号城市驻扎军队时开销最小。
- 对于第二个要求, 在 1 号、2 号、3 号城市驻扎军队时开销最小。
- 第三个要求是无法满足的, 因为在 1 号、5 号城市都不驻扎军队就意味着由道路直接连接的两座城市中都没有驻扎军队。

数据规模与约定

测试点编号	type	$n = m =$
1, 2	A3	10
3, 4	C3	10
5, 6	A3	100
7	C3	100
8, 9	A3	2×10^3
10, 11	C3	2×10^3
12, 13	A1	10^5
14, 15, 16	A2	10^5
17	A3	10^5
18, 19	B1	10^5
20, 21	C1	10^5
22	C2	10^5
23, 24, 25	C3	10^5

数据类型的含义：

- A：城市 i 与城市 $i + 1$ 直接相连。
- B：任意城市与城市 1 的距离不超过 100（距离定义为最短路径上边的数量），即如果这棵树以 1 号城市为根，深度不超过 100。
- C：在树的形态上无特殊约束。
- 1：询问时保证 $a = 1, x = 1$ ，即要求在城市 1 驻军。对 b, y 没有限制。
- 2：询问时保证 a, b 是相邻的（由一条道路直接连通）
- 3：在询问上无特殊约束。

对于 100% 的数据，保证 $1 \leq n, m \leq 10^5, 1 \leq p_i \leq 10^5, 1 \leq u, v, a, b \leq n, a \neq b, x, y \in \{0, 1\}$ 。