

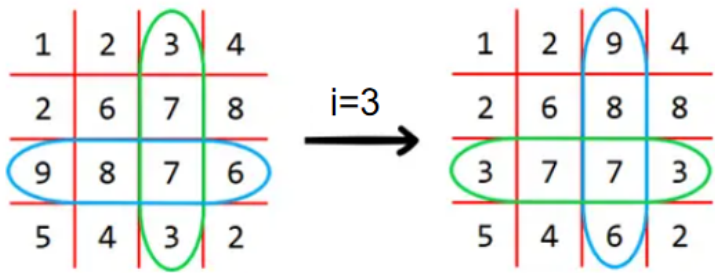
matrix

1.0 秒    512 MiB

【题目描述】

给定一个  $n \times n$  的矩阵  $A$ ，定义一次操作为选择一个整数  $i$ ，对于所有整数  $j \in [1, n]$  且  $j \neq i$ ，交换  $A_{j,i}$  与  $A_{i,j}$ 。

例如：当  $n = 4, i = 3$  时，这个矩阵  $A$  会发生如下交换：



求出在进行若干次操作后可达成的矩阵中，以  $A_{1,1}, A_{1,2}, \dots, A_{1,n}, A_{2,1}, A_{2,2}, \dots, A_{n,n}$  的形式排列后，字典序最小的矩阵。

【输入格式】

从文件 `matrix.in` 中读入数据。  
第一行一个整数  $n$  代表矩阵大小。  
接下来  $n$  行每行  $n$  个整数，代表  $A_{i,j}$ 。

【输出格式】

输出到文件 `matrix.out` 中。  
输出共  $n$  行，每行  $n$  个整数，表示可达成的字典序最小的矩阵。

【样例 1 输入】

```
1 3
2 2 1 2
3 2 1 2
4 1 1 2
```

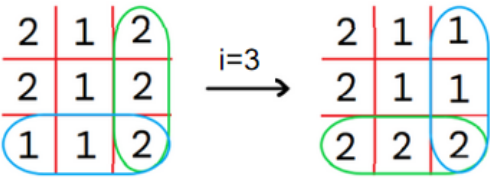
【样例 1 输出】

```
1 2 1 1
2 2 1 1
```

3

2 2 2

【样例 1 解释】



【样例 2 输入】

1 4

2 3 3 1 2

3 1 1 3 1

4 3 2 3 2

5 2 3 3 1

【样例 2 输出】

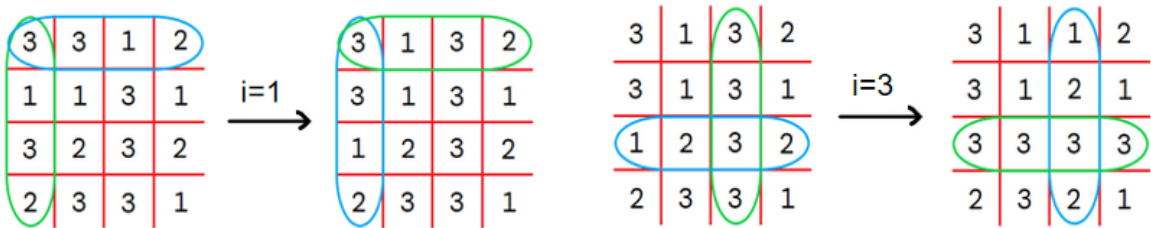
1 3 1 1 2

2 3 1 2 1

3 3 3 3 3

4 2 3 2 1

【样例 2 解释】



【样例 3】

见选手目录下的 *matrix/matrix3.in* 与 *matrix/matrix3.ans*。  
该样例数据满足测试点 5 ~ 8 的性质。

**【样例 4】**

见选手目录下的 *matrix/matrix4.in* 与 *matrix/matrix4.ans*。  
该样例数据满足测试点 9 ~ 12 的性质。

**【样例 5】**

见选手目录下的 *matrix/matrix5.in* 与 *matrix/matrix5.ans*。  
该样例数据满足测试点 13 ~ 20 的性质。

**【数据范围】**

对于 100% 的数据，满足  $n \leq 1000, 1 \leq A_{i,j} \leq 10^9$ 。

测试点编号	$n \leq$
1 ~ 4	5
5 ~ 8	20
9 ~ 12	200
13 ~ 20	1000

## triangle

1.0 秒 512 MiB

## 【题目描述】

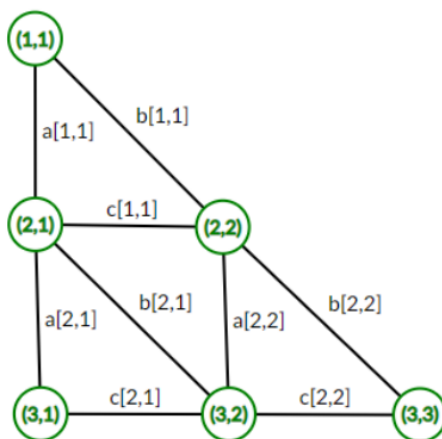
小 X 有一个  $n$  阶三角图。一个  $n$  阶三角图是一个无向带权图，图上有  $\frac{n(n+1)}{2}$  个节点以及  $\frac{3n(n-1)}{2}$  条边。

我们可以把这张图想象为一个大小为  $n \times n$  的二维网格的一个下三角部分，令  $(i, j)$  表示二维网格的  $i$  行  $j$  列，那么所有节点就是满足  $1 \leq j \leq i \leq n$  的点  $(i, j)$ 。

图中有三类边，具体地，对于所有  $1 \leq j \leq i < n$  都有：

- 一条连接  $(i, j)$  和  $(i+1, j)$  的边权为  $a_{i,j}$  的边。
- 一条连接  $(i, j)$  和  $(i+1, j+1)$  的边权为  $b_{i,j}$  的边。
- 一条连接  $(i+1, j)$  和  $(i+1, j+1)$  的边权为  $c_{i,j}$  的边。

如图所示，这是一个 3 阶三角图：



特别地，这张图十分符合三角形的性质，于是对于任意  $1 \leq j \leq i < n$ ，都存在一个边长分别为  $a_{i,j}$ ,  $b_{i,j}$  与  $c_{i,j}$  的三角形。

现在小 X 位于三角图上的  $(1, 1)$  节点，他想到达  $(n, n)$  节点，由于他想要尽可能游览这张图但又不想游览重复的地点，所以他想让你求出从  $(1, 1)$  到  $(n, n)$  的最长路径，使得每条边至多经过一次。

## 【输入格式】

从文件 **triangle.in** 中读入数据。

第一行输入一个整数表示  $n$ 。

接下来  $n-1$  行，其中的第  $i$  行输入  $i$  个整数，第  $i$  行第  $j$  个整数表示  $a_{i,j}$ 。

接下来  $n-1$  行，其中的第  $i$  行输入  $i$  个整数，第  $i$  行第  $j$  个整数表示  $b_{i,j}$ 。

接下来  $n-1$  行，其中的第  $i$  行输入  $i$  个整数，第  $i$  行第  $j$  个整数表示  $c_{i,j}$ 。

【输出格式】

输出到文件 *triangle.out* 中。  
输出一行一个整数表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 38
```

【样例 2 输入】

```
1 5
2 9
3 3 9
4 4 8 7
5 6 6 7 5
6 3
7 9 3
8 7 7 5
9 8 4 6 8
10 9
11 7 5
12 9 8 3
```

13

8 7 6 4

【样例 2 输出】

1

171

【样例 3】

见选手目录下的 *triangle/triangle3.in* 与 *triangle/triangle3.ans*。  
该样例数据满足测试点 7 ~ 10 的性质。

【样例 4】

见选手目录下的 *triangle/triangle4.in* 与 *triangle/triangle4.ans*。  
该样例数据满足测试点 11 ~ 14 的性质。

【样例 5】

见选手目录下的 *triangle/triangle5.in* 与 *triangle/triangle5.ans*。  
该样例数据满足测试点 15 ~ 20 的性质。

【数据范围】

对于 100% 的数据，满足  $1 \leq n \leq 1000$ ，保证  $1 \leq a_{i,j}, b_{i,j}, c_{i,j} \leq 10^9$ ，保证对于任意  $(i, j)$  都有  $a_{i,j} + b_{i,j} > c_{i,j}$  且  $a_{i,j} + c_{i,j} > b_{i,j}$  且  $b_{i,j} + c_{i,j} > a_{i,j}$ 。

测试点编号	$n \leq$
1 ~ 6	5
7 ~ 10	16
11 ~ 14	200
15 ~ 20	1000

## build

1.0 秒 512 MiB

## 【题目描述】

在 C 国有  $n$  个村子，以及  $n - 1$  条待修建的双向公路，第  $i$  条公路连接  $u_i, v_i$ ，修建的代价为  $c_i$ ，保证任意两个村子可以通过一些双向公路互相到达。

已知国王和  $k$  个守卫在某个村子  $r$ ，第  $i$  个守卫需要到达某个村子  $x_i$ 。

国王会选择代价之和尽可能少的一些公路进行修建，使得每个守卫可以从  $r$  出发经过这些公路到达目的地。

对于每个  $r \in [1, n]$ ，求在所有选择  $x_i \in [1, n]$  的情况下，国王修建公路所需代价之和的最大值。

## 【输入格式】

从文件 *build.in* 中读入数据。

第一行，两个正整数  $n, k$ 。

之后  $n - 1$  行，每行两个正整数  $u_i, v_i$  和一个非负整数  $c_i$ 。

## 【输出格式】

输出到文件 *build.out* 中。

输出共  $n$  行，第  $r$  行一个非负整数，表示对应的答案。

## 【样例 1 输入】

```
1 11 3
2 1 2 5
3 2 3 3
4 2 6 5
5 3 4 4
6 3 5 2
7 1 7 6
8 7 8 4
9 7 9 5
10 1 10 1
11 10 11 1
```

## 【样例 1 输出】

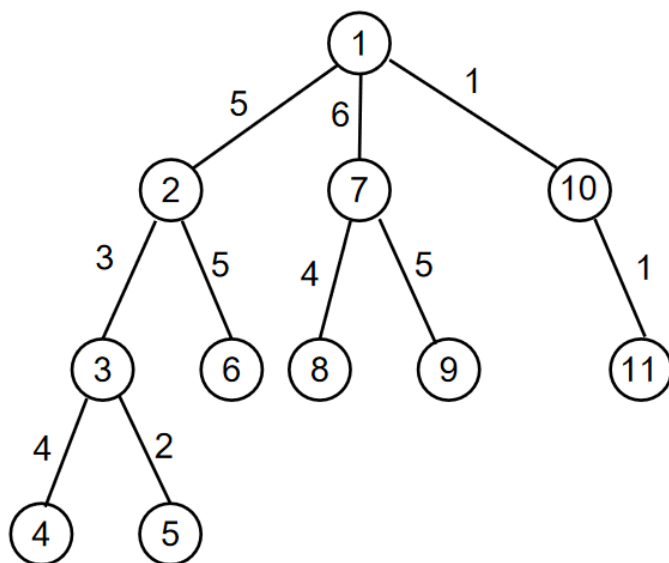
```

1 28
2 28
3 28
4 32
5 30
6 32
7 28
8 32
9 32
10 29
11 30

```

### 【样例 1 解释】

例如对于  $r = 1$  的情况，当  $x_1 = 4$ ,  $x_2 = 6$ ,  $x_3 = 9$  时国王需要修建代价之和为 28 的公路，且可以证明不会有代价更大的情况。



### 【样例 2】

见选手目录下的 *build/build2.in* 与 *build/build2.ans*。  
该样例满足测试点 3 ~ 5 的限制。

### 【样例 3】

见选手目录下的 *build/build3.in* 与 *build/build3.ans*。



该样例满足测试点 6 ~ 9 的限制。

【样例 4】

见选手目录下的 *build/build4.in* 与 *build/build4.ans*。  
该样例满足测试点 10 ~ 14 的限制。

【样例 5】

见选手目录下的 *build/build5.in* 与 *build/build5.ans*。  
该样例满足测试点 15 ~ 18 的限制。

【样例 6】

见选手目录下的 *build/build6.in* 与 *build/build6.ans*。  
该样例满足测试点 19 ~ 25 的限制。

【数据范围】

对于 100% 的数据，满足  $1 \leq k \leq n \leq 10^5$ ， $0 \leq c_i \leq 10^9$ 。

测试点编号	$n \leq$	$k$
1 ~ 2	18	$\leq 10^5$
3 ~ 5	200	$\leq 20$
6 ~ 9	1000	$\leq 100$
10 ~ 14	2000	$\leq 10^5$
15 ~ 18	$10^5$	$= 1$
19 ~ 25		$\leq 10^5$

war

2.0 秒 512 MiB

**【题目描述】**

给定一棵以 1 为根的含  $n$  个节点的树，所有边的初始权值为 0。

有  $m$  次操作，每次操作可以写为  $1\ x\ y$  或者  $2\ i$  的形式，前者表示  $x$  节点到  $y$  节点的路径上的所有边权值增加 1，后者表示撤销第  $i$  次操作（保证第  $i$  次操作为增加操作且尚未被撤销）。

接下来有  $q$  次询问，每次询问可以写为  $l\ r$  的形式，表示询问满足第  $l$  次操作后到第  $r$  次操作后权值始终不为 0 的深度最大的边（定义根节点的深度为 0，边的深度定义为连接的两个点的深度最大值）。

**【输入格式】**

从文件 **war.in** 中读入数据。

第一行三个整数  $n, m, q$ ，分别代表树的节点数，操作次数，询问次数。

接下来  $n - 1$  行，每行两个整数  $u, v$ ，代表树上存在一条从  $u$  节点到  $v$  节点的边，保证所有边形成一棵树。

接下来  $m$  行，每行首先一个整数  $op$ ，如果  $op = 1$  则紧接着有两个整数  $x, y$ ，否则如果  $op = 2$  则紧接着有一个整数  $i$ ，代表一次操作。

接下来  $q$  行，每行两个整数  $l, r$ ，代表一次询问。

**【输出格式】**

输出到文件 **war.out** 中。

输出  $q$  行，每行一个整数代表该次询问的答案。

**【样例 1 输入】**

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

11	1	3
12	2	5
13	3	3
14	1	2
15	4	5

【样例 1 输出】

1	1
2	2
3	2
4	1
5	2

【样例 1 解释】

每条边的深度及每个时刻的权值如下：

1 − 2 : 1 2 3 2 1，深度为 1。

2 − 3 : 0 1 2 2 1，深度为 2。

2 − 4 : 0 0 1 1 0，深度为 2。

1 − 5 : 0 1 1 1 1，深度为 1。

因此询问区间 [1, 3], [2, 5], [3, 3], [1, 2], [4, 5] 的答案分别为 1, 2, 2, 1, 2。

【样例 2 输入】

1	6	10	10
2	5	1	
3	1	6	
4	6	3	
5	5	4	
6	6	2	
7	1	5	1
8	1	6	2
9	1	6	1
10	1	5	3
11	1	6	4
12	1	4	2

```
13 1 4 1
14 2 5
15 2 6
16 2 4
17 4 7
18 1 3
19 3 3
20 1 7
21 1 4
22 4 5
23 3 4
24 1 1
25 1 1
26 1 5
```

**【样例 2 输出】**

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

**【样例 3】**

见选手目录下的 *war/war3.in* 与 *war/war3.ans*。  
该样例满足测试点 1 ~ 4 的限制。

**【样例 4】**

见选手目录下的 *war/war4.in* 与 *war/war4.ans*。  
该样例满足测试点 5 ~ 10 的限制。

【样例 5】

见选手目录下的 *war/war5.in* 与 *war/war5.ans*。  
该样例满足测试点 11 ~ 16 的限制。

【样例 6】

见选手目录下的 *war/war6.in* 与 *war/war6.ans*。  
该样例满足测试点 17 ~ 20 的限制。

【数据范围】

对于 100% 的数据，满足  $1 \leq n, m \leq 10^5, 1 \leq q \leq 5 \times 10^5, 1 \leq u, v \leq n, op \in [1, 2], 1 \leq x, y \leq n, x \neq y, 1 \leq i \leq m, 1 \leq l \leq r \leq m$ 。

测试点编号	$n, m \leq$	$q \leq$	特殊性质
1 ~ 4	$5 \times 10^3$	$5 \times 10^3$	无
5 ~ 10	$10^5$	$5 \times 10^5$	A
11 ~ 16	$5 \times 10^4$	$2 \times 10^5$	无
17 ~ 20	$10^5$	$5 \times 10^5$	无

特殊性质 A：满足  $x = 1$ 。