

解释其中的部分询问；下面的解释中用(a,b;t,v)表示在 t 时刻出现的服务器 a 和 b 之间的重要度为 v 的请求：

对于第一个询问（在时刻 1），此时没有任何请求，输出 -1。

对于第四个询问（在时刻 6），此时有两条交互 (8,13;2,3),(9,12;3,5)，所有询问均经过 2 号服务器，输出 -1。

对于第五个询问（在时刻 8），此时有三条交互(8,13;2,3),(9,12;3,5),(10,12;7,1)，只有交互 (10,12;7,1) 没有经过 2 号服务器，因此输出其重要度 1。

对于最后一个询问（在时刻 23），此时有三条交互 (9,5;12,6),(9,12;16,4),(10,5;17,7)。当 3 号服务器出现故障时，只有交互 (9,5;12,6) 没有经过 3 号服务器，因此输出 6。

#### 【数据范围】

对于 20%的数据， $n, m \leq 1000$ 。

对于 30%的数据， $m \leq 2000$ 。

另有 20%的数据，第一次事件必定是加入重要度最大的交互，且没有 type=1 的事件。

另有 20%的数据，树是一条链，且满足 i 与 i+1 之间有边。

对于 100%的数据， $2 \leq n \leq 10^5, 1 \leq m \leq 2 \times 10^5$ ，其他的所有输入值不超过  $10^9$ 。

为了便于你分辨第三类数据（即第一次事件必定加入重要度最大的交互），规定这条交互的重要度为  $10^9$ 。而在其他类型的数据中，任何交互的重要度都小于该值。

#### 【编译命令】

对于c++语言： `g++ -o network network.cpp -lm`

对于c语言： `gcc -o network network.c -lm`

对于pascal语言： `fpc network.pas`

## 第 3 题：树(tree)，时间限制 2s，内存限制 128M。

#### 【问题描述】

小 A 想做一棵很大的树，但是他手上的材料有限，只好用点小技巧了。

开始，小 A 只有一棵结点数为 N 的树，结点的编号为 1,2,...,N，其中结点 1 为根；我们称这颗树为模板树。小 A 决定通过这棵模板树来构建一颗大树。构建过程如下：

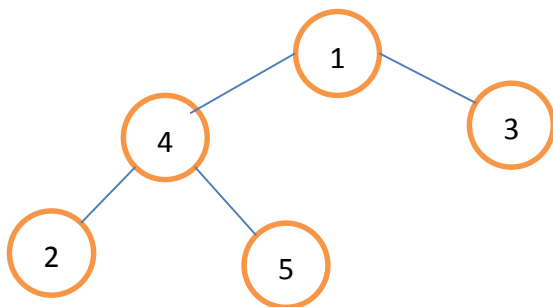
- (1) 将模板树复制为初始的大树。
- (2) 以下(2.1)(2.2)(2.3)步循环执行 M 次

(2.1) 选择两个数字  $a, b$ , 其中  $1 \leq a \leq N$ ,  $1 \leq b \leq$  当前大树的结点数。

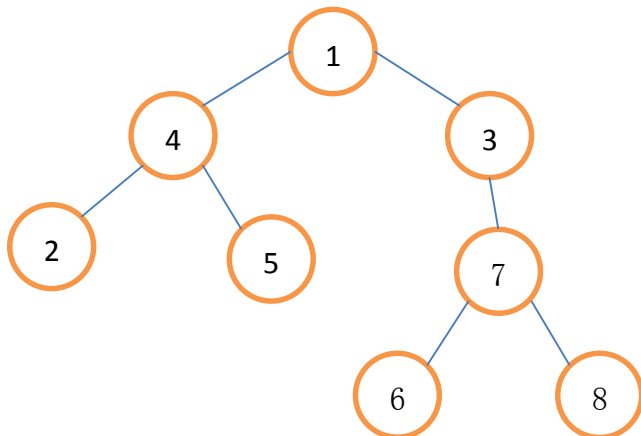
(2.2) 将模板树中以结点  $a$  为根的子树复制一遍, 挂到大树中结点  $b$  的下方(也就是说, 模板树中的结点  $a$  为根的子树复制到大树中后, 将成为大树中结点  $b$  的子树)。

(2.3) 将新加入大树的结点按照在模板树中编号的顺序重新编号。例如, 假设在进行 2.2 步之前大树有  $L$  个结点, 模板树中以  $a$  为根的子树共有  $C$  个结点, 那么新加入模板树的  $C$  个结点在大树中的编号将是  $L+1, L+2, \dots, L+C$ ; 大树中这  $C$  个结点编号的大小顺序和模板树中对应的  $C$  个结点的大小顺序是一致的。

下面给出一个实例。假设模板树如下图:



根据第(1)步, 初始的大树与模板树是相同的。在(2.1)步, 假设选择了  $a=4$ ,  $b=3$ 。运行(2.2)和(2.3)后, 得到新的大树如下图所示:



现在他想问你, 树中一些结点对的距离是多少。

【程序文件名】

源程序文件名为 `tree.cpp/c/pas`。

【输入格式】

输入文件名为 `tree.in`。

第一行三个整数:  $N, M, Q$ , 以空格隔开,  $N$  表示模板树结点数,  $M$  表示第(2)中的循环操作的次数,  $Q$  表示询问数量。

接下来  $N-1$  行, 每行两个整数  $fr, to$ , 表示模板树中的一条树边。

再接下来  $M$  行, 每行两个整数  $x, to$ , 表示将模板树中  $x$  为根的子树复制到大树中成为结点  $to$  的子树的一次操作。

再接下来  $Q$  行, 每行两个整数  $fr, to$ , 表示询问大树中结点  $fr$  和  $to$  之间的距离是多少。

【输出格式】

输出  $Q$  行, 每行一个整数, 第  $i$  行是第  $i$  个询问的答案。

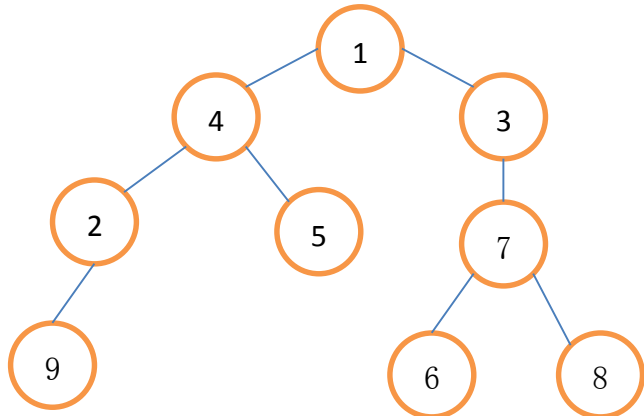
【输入输出样例】

tree.in	tree.out
5 2 3	6

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

【样例解释】

经过两次操作后，大树变成了下图所示的形状：



结点 6 到 9 之间经过了 6 条边，所以距离为 6；类似地，结点 1 到 8 之间经过了 3 条边；结点 5 到 3 之间也经过了 3 条边。

【数据范围】

- 对于 30%的数据， $N,M \leq 500$ ；
- 对于 60%的数据， $N \leq 3000$ ；
- 对于 100%的数据， $N,M,Q \leq 100000$ 。

【编译命令】

- 对于c++语言： `g++ -o tree tree.cpp -lm`
- 对于c语言： `gcc -o tree tree.c -lm`
- 对于pascal语言： `fpc tree.pas`