

天天爱跑步（running）

【问题描述】

小C同学认为跑步非常有趣，于是决定制作一款叫做《天天爱跑步》的游戏。《天天爱跑步》是一个养成类游戏，需要玩家每天按时上线，完成打卡任务。

这个游戏的地图可以看作一棵包含 n 个结点和 $n - 1$ 条边的树，每条边连接两个结点，且任意两个结点存在一条路径互相可达。树上结点编号为从 1 到 n 的连续正整数。

现在有 m 个玩家，第 i 个玩家的起点为 S_i ，终点为 T_i 。每天打卡任务开始时，所有玩家在第 0 秒同时从自己的起点出发，以每秒跑一条边的速度，不间断地沿着最短路径向着自己的终点跑去，跑到终点后该玩家就算完成了打卡任务。（由于地图是一棵树，所以每个人的路径是唯一的）

小C想知道游戏的活跃度，所以在每个结点上都放置了一个观察员。在结点 j 的观察员会选择在第 W_j 秒观察玩家，一个玩家能被这个观察员观察到当且仅当该玩家在第 W_j 秒也正好到达了结点 j 。小C想知道每个观察员会观察到多少人？

注意：我们认为一个玩家到达自己的终点后该玩家就会结束游戏，他不能等待一段时间后再被观察员观察到。即对于把结点 j 作为终点的玩家：若他在第 W_j 秒前到达终点，则在结点 j 的观察员不能观察到该玩家；若他正好在第 W_j 秒到达终点，则在结点 j 的观察员可以观察到这个玩家。

【输入格式】

从文件 `running.in` 中读入数据。

第一行有两个整数 n 和 m 。其中 n 代表树的结点数量，同时也是观察员的数量， m 代表玩家的数量。

接下来 $n - 1$ 行每行两个整数 u 和 v ，表示结点 u 到结点 v 有一条边。

接下来一行 n 个整数，其中第 j 个整数为 W_j ，表示结点 j 出现观察员的时间。

接下来 m 行，每行两个整数 S_i 和 T_i ，表示一个玩家的起点和终点。

对于所有的数据，保证 $1 \leq S_i, T_i \leq n$ ， $0 \leq W_j \leq n$ 。

【输出格式】

输出到文件 `running.out` 中。

输出 1 行 n 个整数，第 j 个整数表示结点 j 的观察员可以观察到多少人。

【样例1输入】

6 3

2 3

```
1 2
1 4
4 5
4 6
0 2 5 1 2 3
1 5
1 3
2 6
```

【样例1输出】

```
2 0 0 1 1 1
```

【样例1说明】

对于 1 号点， $W_1 = 0$ ，故只有起点为 1 号点的玩家才会被观察到，所以玩家 1 和玩家 2 被观察到，共 2 人被观察到。

对于 2 号点，没有玩家在第 2 秒时在此结点，共 0 人被观察到。

对于 3 号点，没有玩家在第 5 秒时在此结点，共 0 人被观察到。

对于 4 号点，玩家 1 被观察到，共 1 人被观察到。

对于 5 号点，玩家 2 被观察到，共 1 人被观察到。

对于 6 号点，玩家 3 被观察到，共 1 人被观察到。

【样例2输入】

```
5 3
1 2
2 3
2 4
1 5
0 1 0 3 0
3 1
1 4
5 5
```

【样例2输出】

```
1 2 1 0 1
```

【子任务】

每个测试点的数据规模及特点如下表所示。提示：数据范围的个位上的数字可以帮助判断是哪一种数据类型。

测试点编号	n	m	约定
1	= 991	= 991	所有人的起点等于自己的终点， 即 $S_i = T_i$
2			
3	= 992	= 992	$W_j = 0$
4			
5	= 993	= 993	无
6	= 99994	= 99994	树退化成一条链，其中1与2有边， 2与3有边， \dots ， $n-1$ 与 n 有边
7			
8			
9	= 99995	= 99995	所有的 $S_i = 1$
10			
11			
12			
13	= 99996	= 99996	所有的 $T_i = 1$
14			
15			
16			
17	= 99997	= 99997	无
18			
19			
20	= 299998	= 299998	

【提示】

如果你的程序需要用到较大的栈空间（这通常意味着需要较深层数的递归），请务必仔细阅读选手目录下的文档`running/stack.pdf`，以了解在最终评测时栈空间的限制与在当前工作环境下调整栈空间限制的方法。