天天爱跑步

【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,，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?
碧丄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 c 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】
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 $1 < S_i, T_i < n$，$0 < W_j < n$ -
【输出格式】
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
4
4 5
4 6
0 2 5 1 2 3
15
13
26
```
【样例 1 输出】
```
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 1 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。

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

在最终评测时，调用栈占用的空间大小不会有单独的限制，但在我们的工作 环境中默认会有 8 MB 的限制。这可能会引起函数调用层数较多时，程序发生 栈溢出崩溃。

我们可以使用一些方法修改调用栈的大小限制。例如，在终端中输入下列命 令

```
ulimit -s 1048576
```

此命令的意义是，将调用栈的大小限制修改为 1 GB。

例如，在选手目录建立如下 *sample.cpp* 或 *sample.pas*

sample.cpp	*sample.pas*
void dfs(int a){	procedure dfs(a: longint);
if(a == 0)	var t: longint;
return;	begin
int t = a;	if a = 0 then
dfs(a - 1);	exit;
}	t := a;
int main(){	dfs(a - 1);
dfs(1000000);	end;
return 0;	begin
}	dfs(1000000);

end.

将上述源代码编译为可执行文件 *sample* 后，可以在终端中运行如下命令运 行该程序

./sample

如果在没有使用命令“ulimit -s 104857 6"的情况下运行该程序,*sample* 会因为栈溢出而崩溃；如果使用了上述命令后运行该程序，该程序则不会崩溃。

特别地，当你打开多个终端时，它们并不会共享该命令，你需要分别对它们 运行该命令。

请注意，**调用栈占用的空间会计入总空间占用中，和程序其他部分占用的内 存共同受到内存限制**。