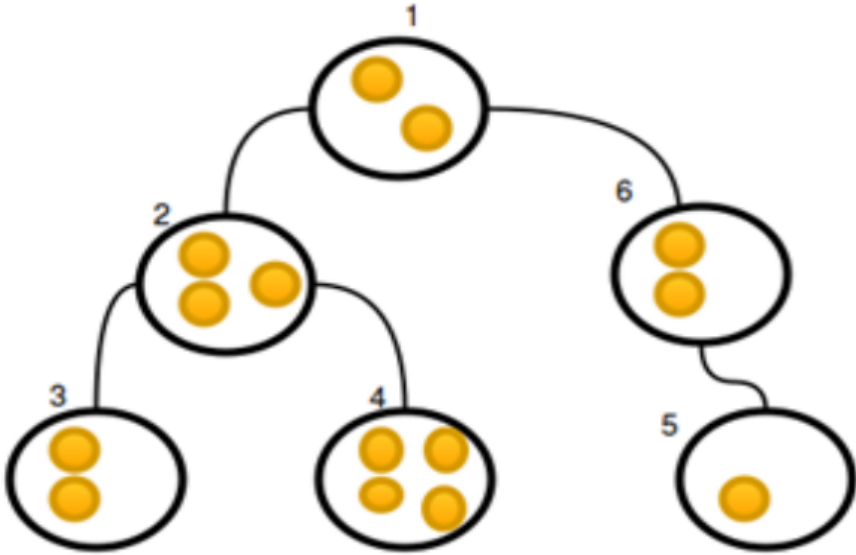


Nik and Ann are playing a game. They have a tree with  $n$  nodes numbered from 1 to  $n$ . Each node has 0 to  $m$  ( $0 \leq m \leq 10^9$ ) golden coins.



Before the game starts they select a single node as the **treasure chest**. Suppose  $d(u)$  denotes the distance between node  $u$  and the treasure chest.

The players make their move alternatively. In each move they do the followings:

- Select any single node  $u$  which has positive number of coins in it.
- Pick one coin from that node and move it to any node  $v$  such that  $d(v) < d(u)$ .

The game ends when no one can make a move (when all the coins are in the chest). Whoever can't make a move loses the game. Ann always makes the first move and both play optimally.

As Nik is a good programmer, he knows that if the treasure chest is chosen carefully, he can always win the game.

Find number of ways the treasure chest can be chosen so that Nik always wins the game.

### Input

First line will contain an integer number  $T$  ( $1 \leq T \leq 100$ ) denoting number of test cases. First line of each test case will consist of a single integer  $n$  ( $1 \leq n \leq 1000$ ). Next line will contain  $n$  integers, where  $i$ -th integer denotes number of coins in node  $i$ . Each of the next  $n-1$  lines will contain two integer numbers  $u$  and  $v$  denoting an edge of the tree.

### Output

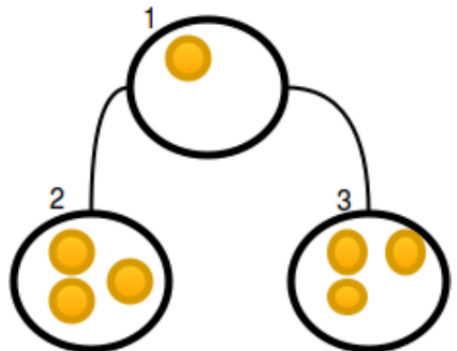
For each case, print case number and number of ways the treasure chest can be chosen so that Nik always wins the game.

Sample input looks like this:

Nik can win only if node **1** is chosen as treasure chest.

### Sample Input

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



### Sample Output

Case 1: 1