

## Problem I. 01tree

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 1024 mebibytes

There is a tree with  $n$  nodes. Every node has a value of 0 or 1.

In one second, you can choose two adjacent nodes with the same value and flip both values.

Given some starting state and some ending state, you will always spend the least number of seconds transforming the starting state into the ending state. If it is impossible to transform the starting state into the ending state, you just skip it (so you spend 0 seconds).

The issue is that, for some nodes, you do not remember the value on them (in either the starting state, the ending state, or both). Over all pairs of (starting state, ending state) that are consistent with your memory, find the total amount of time that it will take to transform from the starting state to the ending state. Print this value modulo  $10^9 + 7$ .

### Input

The first line contains an integer  $t$ , the number of test cases ( $1 \leq t \leq 1000$ ). The test cases follow.

The first line of each test case contains one integer  $n$  ( $2 \leq n \leq 10^5$ ) denoting the size of the tree.

Then  $n - 1$  lines follow, each containing two integers  $u$  and  $v$ , denoting the edge that connects  $u$  and  $v$  in the tree.

The following line contains a string  $s$  of length  $n$  consisting of characters "0", "1", and "?". This string denotes your memory of the starting state: "0" and "1" represent the value of the node, and "?" represents that you do not remember the value of the node.

The following line contains a string  $t$  of length  $n$  denoting your memory of the ending state. It follows the same format as the starting state.

It is guaranteed that the sum of  $n$  over all test cases does not exceed  $10^5$ .

### Output

For each test case, print a line with a single integer: the answer modulo  $10^9 + 7$ .

### Example

<i>standard input</i>	<i>standard output</i>
3	1
2	16
1 2	1
00	
11	
3	
1 2	
2 3	
???	
???	
3	
1 2	
2 3	
??1	
0?0	