Balanced Tree

Time Limit: 2 seconds
Memory limit: 512 MB

A D-Balanced Tree (D being a positive integer) is a tree that satisfies the following conditions:

- Each node is either black or white.
- For every black node, there is at least one other black node at distance at most D.
- For every white node, there is at least one other white node at distance at most D.

You are given a tree that may not have the colour of every node decided yet. You have to choose the colour of all remaining nodes in order to minimize the value of D. However, there may be no valid positive integer D such that the tree is D-Balanced (see example).

Full score will be given only if you found a valid coloring that leads to your answer. However, partial score may be given otherwise. It will be required to solve the problem for several trees.

Input

On the first line of the standard input there is a positive integer T representing the number of test cases that will follow. Each test case describes a tree and consists of :

- The first line contains a single integer N, the number of nodes in the tree.
- Each of the following (N-1) lines contain a pair (x, y), representing an edge between x and y.
- The last line contains N values c_i , representing the colour of each node.

Depending on c_i , the color of a node can be :

- white if c_i equals 0
- black if c_i equals 1
- either black or white (you have to choose) if c_i equals -1

Output

The output should contain the answer for each test case on different lines.
Each answer is structured as follows :

- On the first line, print the minimum D value such that the tree is D-Balanced. If there is no such D, you have to print '-1' (without quotation marks) and skip the next line.
- On the second line, if there is any valid positive integer D, you have to print a coloring of the nodes that leads to that answer. If any of the initial restrictions is not met (by overwriting a fixed colour), the answer will be considered wrong. However, if the provided coloring does not lead to the printed D, you may still get partial credit (see scoring).

Restrictions

- The distance between 2 nodes A and B is equal to the number of edges on the unique path that starts in A and ends in B.
- $3 \leq N \leq 500\ 000$
- The sum of N for all trees in a test is at most `500 000`.

- Each subtask may receive full or partial score according to the 'Scoring' section.
- The nodes are 1-indexed. (i.e. 1, 2, 3, …, N)

Scoring

Points	*Restrictions*
10%	$N \le 17$, $T \le 500$
13%	sum of $N \le 100\ 000$, $c_i \in \{0, 1\}$
26%	sum of $N \le 100\ 000$, all trees in input are paths
45%	sum of $N \le 100\ 000$ If the correct answer is $Q(\neq -1)$ and you gave a solution with $D \in \{Q+1, Q+2, Q+3\}$, you will receive 50% of the points for this subtask
6%	no additional restriction

- If the coloring provided is not consistent with your answer but all the initial conditions are met, you will be given 60% points for each subtask.
- The score for a subtask is the minimum score among the tests that form that subtask.
- Please note that four different scores (except 0) can be received for 4th subtask :
  - 100% if $Q = D$ and the coloring is consistent with the answer
  - 60% if $Q = D$ and the coloring is not consistent with the answer
  - 50% if $Q < D < Q + 4$ and the coloring is consistent with the answer
  - 30% if $Q < D < Q + 4$ and the coloring is not consistent with the answer

Example

standard input	*standard output*	
3 3 1 2 2 3 0 0 -1 4 1 2 2 3 2 4 0 1 0 0 6 1 2 2 3 2 4 4 5 4 6 1 0 0 -1 1 0	1 0 0 0 -1 2 1 0 0 1 1 0