Problem B. Little Worm

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	512 mebibytes

Little Worm is living on a tree. The tree has n vertices (and is a connected, undirected acyclic graph), and Worm occupies the whole path between the vertices a and b.

Worm would like to move to another path – the one between vertices c and d – as it is more sunny there. It is known that the paths $a \leftrightarrow b$ and $c \leftrightarrow d$ have no vertices in common.

To change its position on the tree, Worm can make some moves, which consist of entering a free vertex with Worm's either end. Formally, if Worm is currently occupying a path between x and y, it may choose a new vertex z adjacent to x, which is not on the path $x \leftrightarrow y$. Then Worm frees (stops occupying) y, taking z instead. In a similar way, Worm can choose a vertex z' adjacent to y, free x and occupy z'. After a single move Worm still occupies some path, and its length does not change.

Worm is aiming to get to the path between c and d, but being quite lazy, it doesn't plan for more than $10 \cdot n$ moves. Can you help it reach its goal within that limit?

Input

The first line of input contains the number of test cases z ($1 \leq z \leq 7000$). The test cases follow, each one in the following format:

The first line of a test case contains a single integer n ($4 \leq n \leq 100\,000$) – the number of the vertices of a tree. Each of the following $n - 1$ lines contains two integers u, v ($1 \leq u \neq v \leq n$), describing the endpoints of a single edge.

In the next line two integers a and b ($1 \leq a \neq b \leq n$) are given. These are the endpoints of the path that is Worm's starting position.

The next line contains the endpoints of the path which is Worm's goal, given as two integers c and d ($1 \leq c \neq d \leq n$).

The number of vertices on the path between a and b match the number of vertices on the path between c and d. You may also assume that those two paths have no common vertices.

The sum of all values of n over all test cases does not exceed $1\,000\,000$.

Output

For every test case, if Worm cannot reach its goal in $10 \cdot n$ moves, output -1. Otherwise, output a possible sequence of Worm's moves in two lines: first consisting the number of moves q ($1 \leq q \leq 10 \cdot n$) and the other containing q integers $v_1, v_2, \ldots, v_q$ – the required moves. For $i = 1, 2, \ldots, q$, the value v_i should denote the vertex which is entered by Worm in the i-th move. You may output any correct sequence that moves Worm to the goal and has no more than $10 \cdot n$ moves (in particular, you do not have to minimize the number of moves). Assume that Worm is symmetrical – it can move in both directions and it can enter the goal path facing either side.

Example

standard input	standard output

3
6
1 2
1 3
1 4
4 5
4 6
2 3
5 6
15
1 2
1 6
2 3
2 4
2 5
6 7
6 8
5 9
6 10
9 11
9 12
9 13
12 14
14 15
14 13
3 6
6
1 2
1 3
2 4
4 5
5 6
4 6
3 2
	-1
	7
	15 5 2 1 6 7 3
	3
	2 1 3

Note

We only consider simple paths, i.e. with no vertex appearing twice. It is a known fact that any two vertices in a tree can be connected with exactly one simple path.