Problem M. Binary Search Tree

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

How to check if a tree is a binary search tree?

Someone in a Telegram chat

Binary search tree is a rooted tree, in which:

- each vertex can have at most one left child and at most one right child,
- for each non-leaf vertex x, all vertices in its left subtree are less than x. and all vertices in its right subtree are greater than x.

You are given a tree with n vertices. Can this tree, being rooted at some vertex, be a binary search tree, and if it can, what vertices can be a root?

Input

The first line contains an integer n ($1 \le n \le 500000$) — the number of vertices in the tree.

Each of the next $n - 1$ lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$) — the edges of the tree.

Output

If this tree can't be a binary search tree, output "-1".

Otherwise, output all vertices that can be a root, in increasing order.

Examples

standard input	standard output
3 1 2 2 3	1 2 3
3 1 3 3 2	1
4 1 3 3 2 2 4	-1
4 1 2 1 3 1 4	-1