For the purposes of this challenge, we define a *binary search tree* to be a *binary tree* with the following ordering properties:

- The data value of every node in a node's left subtree is *less than* the data value of that node.
- The data value of every node in a node's right subtree is *greater than* the data value of that node.

Given the root node of a binary tree, can you determine if it's also a binary search tree?
Complete the function in your editor below, which has 1 parameter: a pointer to the root of a binary tree. It must return a *boolean* denoting whether or not the binary tree is a binary search tree. You may have to write one or more helper functions to complete this challenge.

Note: We do not consider a binary tree to be a binary search tree if it contains duplicate values.

Input Format

You are not responsible for reading any input from stdin. Hidden code stubs will assemble a binary tree and pass its root node to your function as an argument.

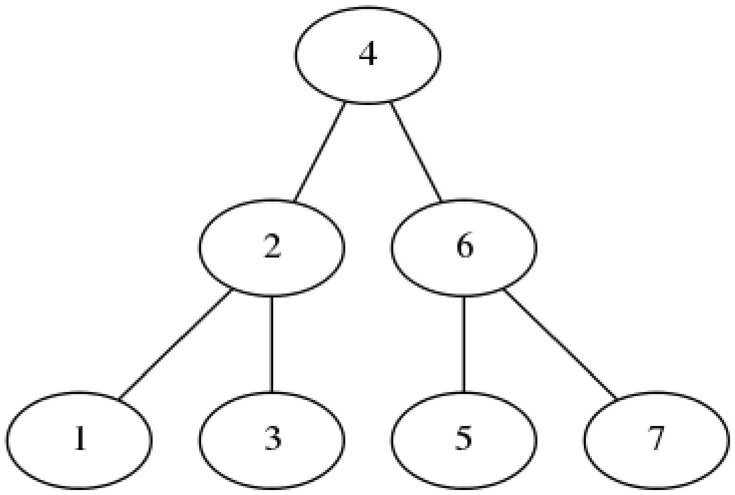
Constraints

- $0 \le data \le 10^4$

Output Format

You are not responsible for printing any output to stdout. Your function must return *true* if the tree is a binary search tree; otherwise, it must return *false*. Hidden code stubs will print this result as a *Yes* or *No* answer on a new line.

Sample Input



Sample Output

```
Yes
```

Explanation

The tree in the diagram satisfies the ordering property for a Binary Search Tree, so we print `Yes`.