

Binary tree traversal and BST check

Rules:

1. Plagiarism is forbidden.
2. Write your program with C++.

Problem Definition:

- ✓ Given preorder traversal and inorder traversal of a binary tree, you will need to find out the postorder traversal of this tree and identify if it is a BST (binary search tree).

I/O Format:

Example:

Input:

5
3 9 20 15 7
9 3 15 20 7

You will first get the number N ($1 \leq N \leq 200$), which is the total number of nodes of the tree. The second and third lines are the tree's **preorder** and **inorder** traversal.

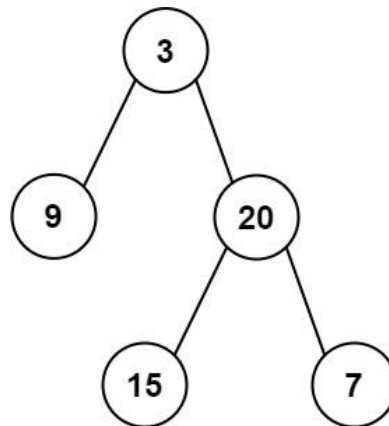
Output:

9 15 7 20 3
No

On the first line, print out the postorder traversal of the tree. And on the second line, if the tree is a BST, print "**Yes**", if not, print "**No**".

Explanation:

✧ The corresponding Tree:



Preorder: 3 9 20 15 7

Inorder: 9 3 15 20 7

Postorder: 9 15 7 20 3

The tree is not a BST because node 9 is in the left subtree of node 3.

Constraints:

- ✓ $1 \leq \text{number of nodes} \leq 200$
- ✓ $1 \leq \text{node value} \leq 200$
- ✓ Each node value is unique
- ✓ Each value that appears in preorder (second line of the input) will also appear in inorder (third line of the input).

Hints:

- ✓ Observing the inorder traversal of the tree, we can find that all the nodes in the left subtree of node 3 are on the left of 3, and all the nodes in the right subtree of node 3 are on the right of 3, so is the node 20.
- ✓ Observing the preorder traversal of the tree, it will first go through the left subtree and then the right subtree. Therefore, build the parent node first then the left subtree, and then the right subtree by using a recursive function.
- ✓ To check if the tree is a BST, try to think about the question below.

