106. Construct Binary Tree from Inorder and Postorder Traversal

Description | Hints | Submissions | Discuss | Solution

Pick One

Given inorder and postorder traversal of a tree, construct the binary tree.

Note:
You may assume that duplicates do not exist in the tree.

For example, given

```
inorder = [9,3,15,20,7]
postorder = [9,15,7,20,3]
```

Return the following binary tree:

```
    3
   / \
  9  20
    /  \
   15   7
```

Seen this question in a real interview before? Yes No

```java
public class L106 {
    public class TreeNode {
            int val;
            TreeNode left;
            TreeNode right;
            TreeNode(int x) { val = x; }
        }

    public TreeNode buildTree(int[] inorder, int[] postorder) {

            if(inorder.length == 0 || postorder.length == 0) {
                return null;
            }
            TreeNode root = new TreeNode(postorder[postorder.length - 1]);
            if(inorder.length == 1 && postorder.length == 1) {
                return root;
            }

            for(int i = 0; i < inorder.length; i ++) {
                if(inorder[i] == postorder[postorder.length - 1]) {
                     root.left = buildTree(assit(inorder, 0, i - 1), assit(postorder, 0, i - 1));
                     root.right = buildTree(assit(inorder, i + 1, inorder.length - 1), assit(postorder, i,
                }
            }
        return root;
      }

    public int [] assit(int [] array, int start, int end) {
        int [] res = new int [end - start + 1];
        for(int i = start; i <= end; i ++) {
            res[i - start] = array[i];
        }
        return res;
    }
}
```