105. Construct Binary Tree from Preorder and Inorder Traversal

Description | Hints | Submissions | Discuss | Solution

Pick One

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

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

For example, given

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

Return the following binary tree:

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

```
/*
 * 先序遍历的第一个结点一定是根节点，查找根节点在中序遍历数组中的位置，前面的节点
 * 便是左子树，后面的便是右子树，然后递归构造完整树。
 */
    public TreeNode buildTree(int[] preorder, int[] inorder) {
            if(preorder.length == 0 || inorder.length == 0)
                return null;

            TreeNode root = new TreeNode(preorder[0]);
            if(preorder.length == 1 && inorder.length == 1)
                return root;

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

    /*
     *  辅助函数，用来切割数组
     */
    public int [] assist(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;
    }
}
```