106. Construct Binary Tree from Inorder and Postorder Traversal

Total Accepted: **55082** Total Submissions: **189297** Difficulty: **Medium**

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

Note:

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

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
// author:zzw
// C++
class Solution {
public:
    TreeNode* buildTree(vector<int>& inorder, vector<int>& postorder) {
        int inorder_size = inorder.size();
        int postorder_size = postorder.size();
        if(inorder_size==0 || postorder_size == 0)
        {
            return NULL;
        }
        return recursionBuild(inorder,0,inorder_size-
1,postorder,0,postorder_size-1);
    }

    TreeNode *recursionBuild(vector<int>& inorder,int inorder_left,int
inorder_right, vector<int>& postorder, int postorder_left, int
postorder_right)
    {
        if(inorder_left>inorder_right)
        {
            return NULL;
        }

        TreeNode *p = new TreeNode(postorder[postorder_right]);
        int i=0;
        for (i=inorder_left;i<inorder_right;i++)
        {
```

```
            if(p->val == inorder[i])
            {
                break;
            }
        }
        p->left = recursionBuild(inorder,inorder_left,i-
1,postorder,postorder_left,postorder_left+i-inorder_left-1);
        p->right
=recursionBuild(inorder,i+1,inorder_right,postorder,postorder_left+i-
inorder_left,postorder_right-1);

        return p;
    }
};
```