106. Construct Binary Tree from Inorder and Postorder Traversal

106 Construct Binary Tree from Inorder and Postorder Traversal

Description

> Given inorder and postorder traversal of a tree, construct the binary tree.
>
> **Note:**
> You may assume that duplicates do not exist in the tree.

Solution

- **Tree**
- **Depth-first Search**
- **Array**

Depth-first Search

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {

public:
    TreeNode* buildTree(vector<int>& inorder, vector<int>& postorder){
    return helper(inorder,0,inorder.size(),postorder,0,postorder.size());
    }

private:
    TreeNode* helper(vector<int>& inorder,int i,int j,vector<int>& postorder,int ii,int jj)
    {
        // 每次取postorder的最后一个值mid，将其作为树的根节点
        // 然后从inroder中找到mid，将其分割成为两部分，左边作为mid的左子树，右边
        // 作为mid的右子树
        // tree:     8 4 10 3 6 9 11
        // Inorder   [3 4 6] 8 [9 10 11]
        // postorder [3 6 4]   [9 11 10] 8

        if(i >= j || ii >= jj)
            return NULL;

        int mid = postorder[jj - 1];

        auto f = find(inorder.begin() + i,inorder.begin() + j,mid);

        int dis = f - inorder.begin() - i;

        TreeNode* root = new TreeNode(mid);
```

```
        root -> left = helper(inorder,i,i + dis,postorder,ii,ii + dis);
        root -> right = helper(inorder,i + dis + 1,j,
            postorder,ii + dis,jj - 1);

        return root;

    }
};
```