一、题目说明

题目105. Construct Binary Tree from Preorder and Inorder Traversal，给二叉树的前序和中序遍历序列，构造一棵二叉树。题目难度是Medium！

二、我的解答

这个题目数据结构上面也有讲，这里用递归遍历算法。前序遍历第1个为树的根，然后用根将中序遍历分成左右子树，再递归就可以了。

代码如下：

```
class Solution{
    public:
        TreeNode* createTree(vector<int> & preorder,vector<int>& inorder,int
leftStart,int leftEnd){
            //preorder中当前元素为树根
            TreeNode* r = new TreeNode(preorder[curRoot]);

            int k=leftStart;
            while(inorder[k] !=preorder[curRoot]) k++;

            curRoot++;
            if(k>leftStart){
                r->left = createTree(preorder,inorder,leftStart,k-1);
            }
            if(k<leftEnd){
                r->right = createTree(preorder,inorder,k+1,leftEnd);
            }
            return r;
        }
        TreeNode* buildTree(vector<int>& preorder,vector<int>& inorder){
            int len = preorder.size();
            if(preorder.size()<1) return NULL;
            TreeNode* root = new TreeNode(preorder[0]);
            if(len==1) return root;

            int k=0;
            while(inorder[k] !=preorder[0]) k++;
            curRoot = 1;
            if(k>0){
                root->left = createTree(preorder,inorder,0,k-1);
            }
            if(k<len-1){
                root->right = createTree(preorder,inorder,k+1,len-1);
            }

            return root;
        }
    private:
        int curRoot;
};
```

性能如下：

```
Runtime: 24 ms, faster than 44.15% of C++ online submissions for Construct
Binary Tree from Preorder and Inorder Traversal.
Memory Usage: 22.2 MB, less than 9.52% of C++ online submissions for Construct
Binary Tree from Preorder and Inorder Traversal.
```

三、优化措施

代码简化如下：

```
class Solution{
    public:
        TreeNode* createTree(vector<int> & preorder,vector<int>& inorder,int
leftStart,int leftEnd){
            TreeNode* r = new TreeNode(preorder[curRoot]);

            int k=leftStart;
            while(inorder[k] !=preorder[curRoot]) k++;

            curRoot++;
            if(k>leftStart){
                r->left = createTree(preorder,inorder,leftStart,k-1);
            }
            if(k<leftEnd){
                r->right = createTree(preorder,inorder,k+1,leftEnd);
            }
            return r;
        }
        TreeNode* buildTree(vector<int>& preorder,vector<int>& inorder){
            int len = preorder.size();
            if(len<1) return NULL;
            curRoot = 0;
            return createTree(preorder,inorder,0,len-1);
        }
    private:
        int curRoot;
};
```

```
Runtime: 20 ms, faster than 64.01% of C++ online submissions for Construct
Binary Tree from Preorder and Inorder Traversal.
Memory Usage: 22.4 MB, less than 9.52% of C++ online submissions for Construct
Binary Tree from Preorder and Inorder Traversal.
```