leetcode笔记

暂时没用到的二级标题

暂时没用到的三级标题

1.dasda

二叉树

145. 二叉树的后序遍历

给你一棵二叉树的根节点 root ， 返回其节点值的后序遍历 。
前中后序全解析

```cpp
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(righ
 * };
 */
class Solution {
public:
    vector<int> postorderTraversal(TreeNode* root) {
        vector<int> res;//配合收藏夹内题解去使用
        stack<TreeNode*> stk;
        if(!root)
            return res;
        stk.emplace(root);
        while(!stk.empty()){//后序遍历即左右中，所以这里要按照
            auto node = stk.top();//中空右左去push
            if(node){
                stk.pop();//我们已经在初始，或者上一次循环push进了中节点，
                //这次循环还会push所以要先pop避免重复
                stk.emplace(node);
                stk.emplace(nullptr);//这个标记一定是放在中节点之前，因为每个节点都有当作中节点的
                //实质上对于递归也是每次只处理中节点，所以我们在遍历的时候在中节点前做标记即可
                if(node->right) stk.emplace(node->right);
                if(node->left) stk.emplace(node->left);
            }
            else{
                stk.pop();
                node = stk.top();
                stk.pop();
                res.push_back(node->val);
            }
        }
        return res;
    }
};
```