106. Construct Binary Tree from Inorder and Postorder Traversal <medium>

class Solution {

int post\_idx = 0;

unordered\_map<int,int> inorder\_map;

public:

TreeNode\* buildTree(vector<int>& inorder, vector<int>& postorder) {

for(int i = 0; i < inorder.size(); i++){

inorder\_map[inorder[i]] = i;

}

post\_idx = postorder.size()-1;

return recursive(postorder, 0, inorder.size()-1);

}

TreeNode\* recursive(vector<int>& postorder, int left\_idx, int right\_idx){

if(left\_idx > right\_idx)

return nullptr;

int root\_val = postorder[post\_idx];

TreeNode\* root = new TreeNode(root\_val);

int index = inorder\_map[root\_val];

post\_idx--;

root->right = recursive(postorder, index + 1, right\_idx);

root->left = recursive(postorder, left\_idx, index - 1);

return root;

}

};