105. Construct Binary Tree from Preorder and Inorder Traversal <medium>

class Solution {

int pre\_idx = 0;

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

public:

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

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

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

}

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

}

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

if(left\_idx > right\_idx)

return nullptr;

int root\_val = preorder[pre\_idx];

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

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

pre\_idx++;

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

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

return root;

}

};