Day 46

Code /\*\*

\* 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(right) {}

\* };

\*/

class Solution {

//FILLING MAP FOR INORDER

void fillmap(map<int ,int>&mp , vector<int>&inorder ,int n){

for(int i = 0 ; i < n ; i ++){

mp[inorder[i]] = i;

}

}

TreeNode\* solve(vector<int>&pre , vector<int>&in , int &pi , int is ,int ie , int n , map<int ,int>&mp)

{

if(pi >= n || (is > ie)){

return NULL;

}

int element = pre[pi++];

TreeNode\* root = new TreeNode(element);

int position = mp[element];

root->left = solve(pre, in , pi , is , position - 1 , n , mp);

root->right = solve(pre, in , pi , position + 1 , ie, n , mp);

return root;

}

public:

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

TreeNode\* ans;

int n = preorder.size();

map<int , int>mp;

fillmap(mp , inorder , n);

int preIndex = 0;

return solve(preorder , inorder , preIndex , 0 , n - 1 , n , mp);

}

};

----------------------------------------------------------------------------------------------------------

Time complexity – O(n)

Space complexity- O(n)

------------------------------------------------------------------------------------------------------------------------------------

