Practical – 23

AIM: Construct Binary Tree from Preorder and Inorder Traversal Given two integer arrays preorder and inorder where preorder is the preorder traversal of a binary tree and inorder is the inorder traversal of the same tree, construct and return the binary tree.

Example 1: Input: preorder = [3,9,20,15,7],

inorder = [9,3,15,20,7]

Output: [3,9,20,null,null,15,7]

* Program

class Solution {

public:

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

int preStart = 0, preEnd = preorder.size() - 1;

int inStart = 0, inEnd = inorder.size() - 1;

map < int, int > mp;

for (int i = inStart; i <= inEnd; i++) {

mp[inorder[i]] = i;

}

TreeNode\* root = constructTree(preorder, preStart, preEnd, inorder, inStart, inEnd, mp);

return root;

}

TreeNode\* constructTree(vector < int > & preorder, int preStart, int preEnd, vector < int > & inorder, int inStart, int inEnd, map < int, int > & mp) {

if (preStart > preEnd || inStart > inEnd) return NULL;

TreeNode\* root = new TreeNode(preorder[preStart]);

int elem = mp[root -> val];

int nElem = elem - inStart;

root -> left = constructTree(preorder, preStart + 1, preStart + nElem, inorder,

inStart, elem - 1, mp);

root -> right = constructTree(preorder, preStart + nElem + 1, preEnd, inorder,

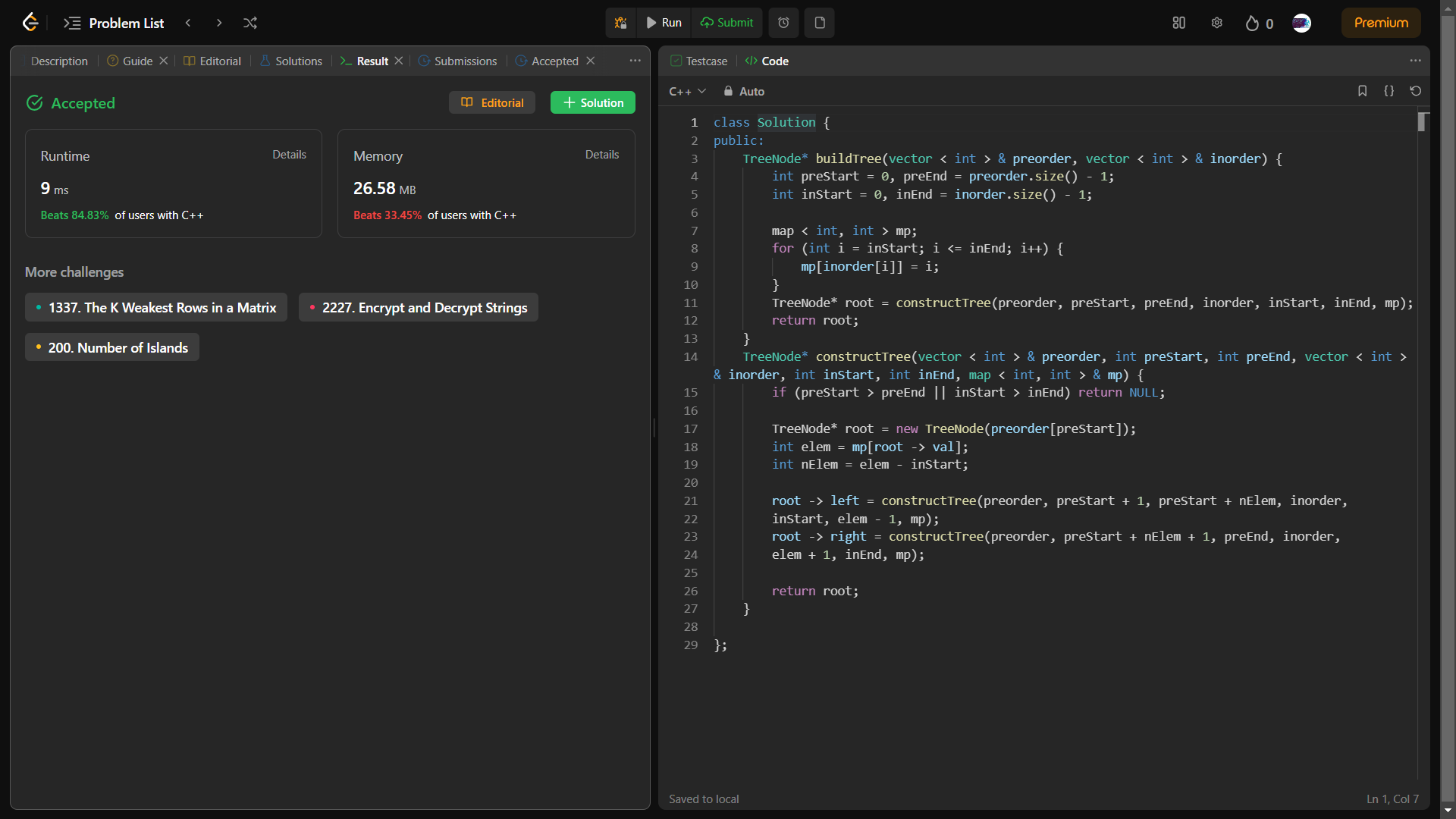
elem + 1, inEnd, mp);

return root;

}

};

* Output



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

Student Signature Faculty Signature Marks