Practical – 24

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

* Program

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

unordered\_map<int, int> inToIndex;

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

inToIndex[inorder[i]] = i;

return build(inorder, 0, inorder.size() - 1, postorder, 0,

postorder.size() - 1, inToIndex);

}

TreeNode\* build(const vector<int>& inorder, int inStart, int inEnd,

const vector<int>& postorder, int postStart, int postEnd,

const unordered\_map<int, int>& inToIndex) {

if (inStart > inEnd)

return nullptr;

const int rootVal = postorder[postEnd];

const int rootInIndex = inToIndex.at(rootVal);

const int leftSize = rootInIndex - inStart;

TreeNode\* root = new TreeNode(rootVal);

root->left = build(inorder, inStart, rootInIndex - 1, postorder, postStart,

postStart + leftSize - 1, inToIndex);

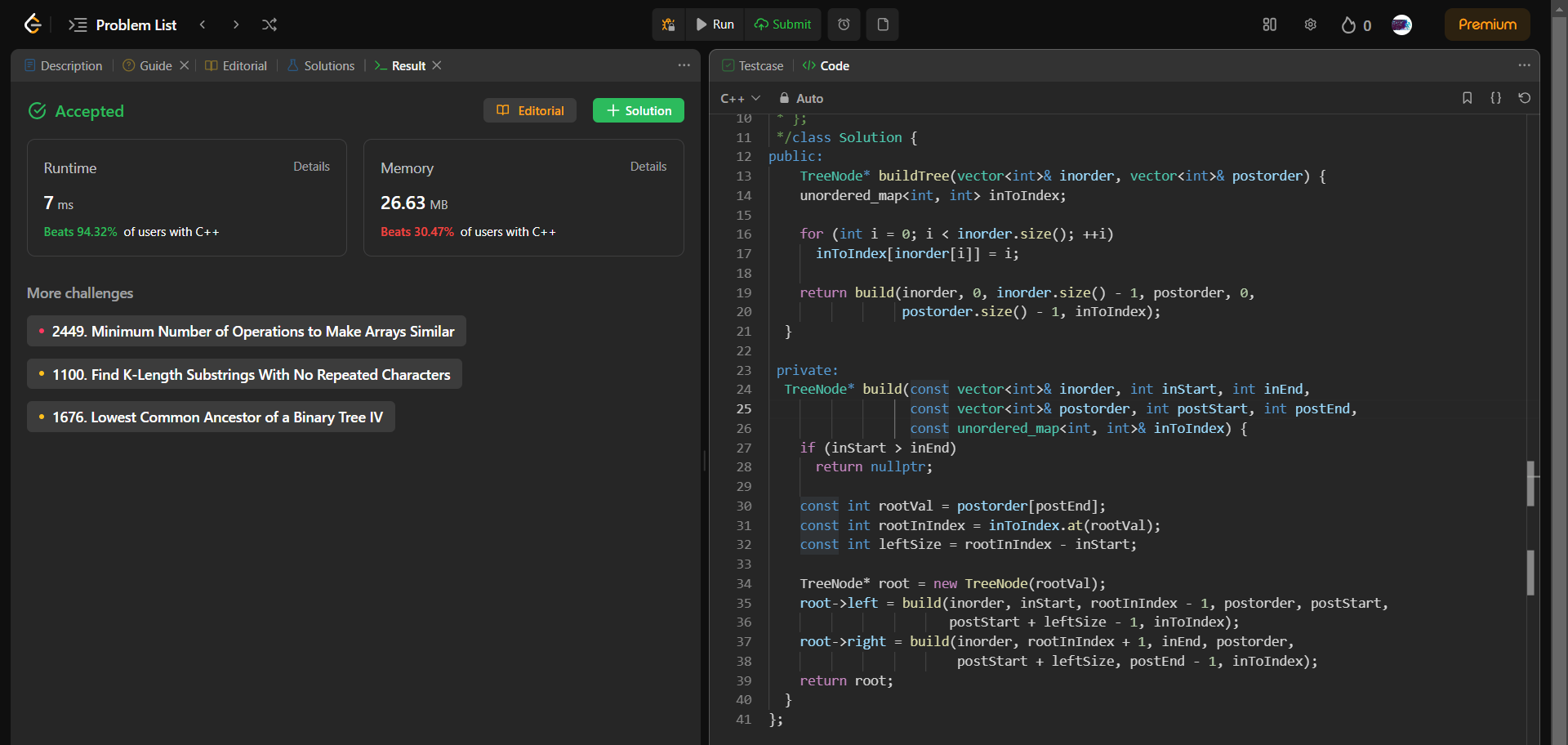
root->right = build(inorder, rootInIndex + 1, inEnd, postorder,

postStart + leftSize, postEnd - 1, inToIndex);

return root;

}

* Output



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

Student Signature Faculty Signature Marks