**Experiment 5.1**

**Student Name:** Sahil Kaundal **UID:** 21BCS8197

**Branch:** BE CSE (Lateral Entry) **Section/Group:** 616/A

**Semester:** 6th **Date of Performance:** 22/03/2023

**Subject Name:** CC-2 Lab **Subject Code:** 20CSP-351

1. **Aim/Overview of the practical:**

Same Tree

Given the roots of two binary trees p and q, write a function to check if they are the same or not. Two binary trees are considered the same if they are structurally identical, and the nodes have the same value.

<https://leetcode.com/problems/same-tree/>

1. **Apparatus / Simulator Used:**

* Windows 7 or above
* Google Chrome

1. **Objective:**

* To understand the concept of Trees.

A tree is a collection of entities called nodes. Nodes are connected by edges. Each node contains a value or data, and it may or may not have a child node. The first node of the tree is called the root.

1. **Code:**

class Solution {

  public boolean isSameTree(TreeNode p, TreeNode q) {

    if (p == null && q == null) {

      return true;

    }

    if (p == null || q == null) {

      return false;

    }

    if (p.val != q.val) {

      return false;

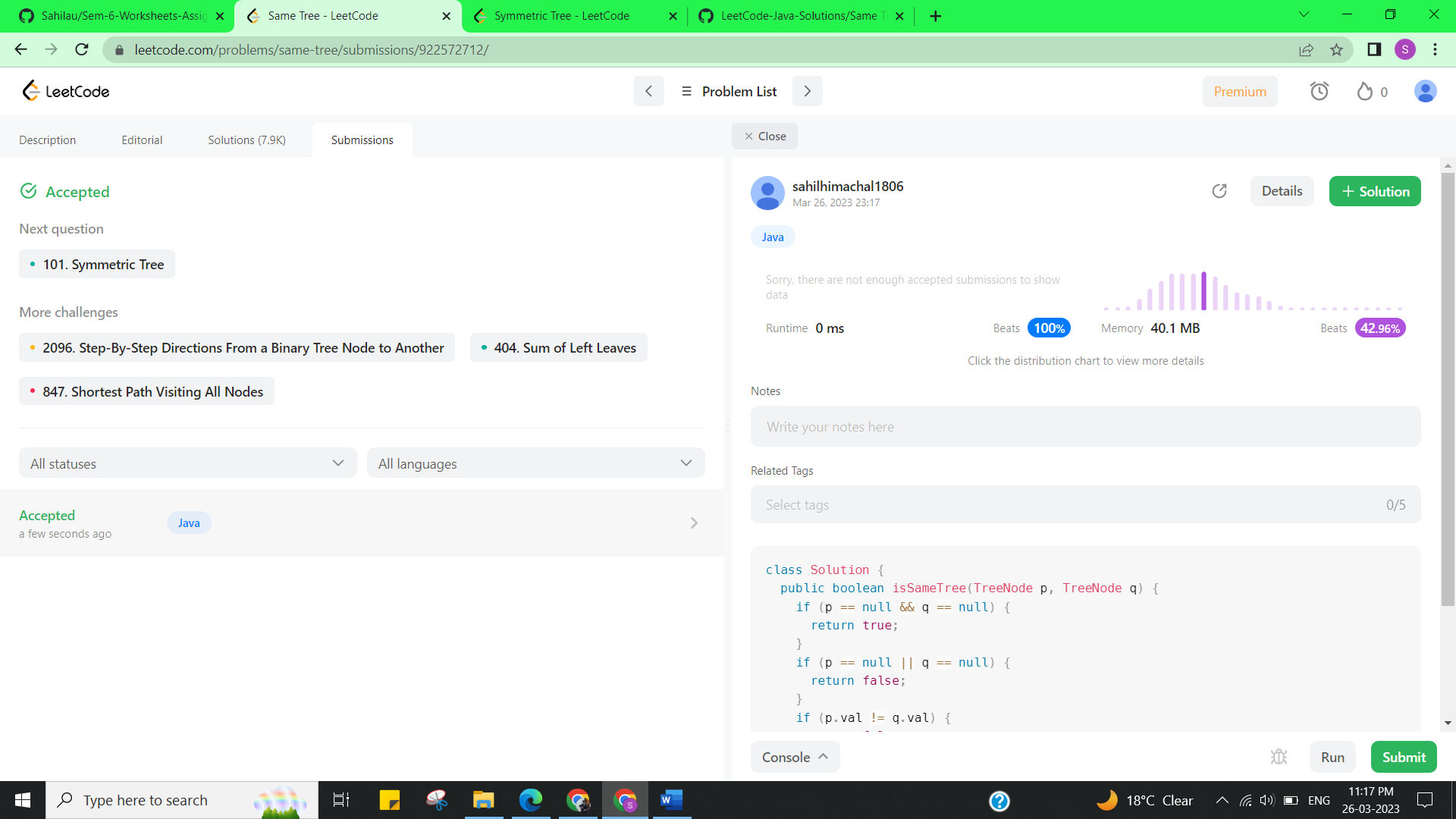
    }

    return isSameTree(p.left, q.left) && isSameTree(p.right, q.right);

  }

}

**4. Result/Output/Writing Summary:**



**Experiment 5.2**

1. **Aim/Overview of the practical:**

Symmetric Tree

Given the root of a binary tree, check whether it is a mirror of itself (i.e., symmetric around its center).

<https://leetcode.com/problems/symmetric-tree/>

1. **Apparatus / Simulator Used:**

* Windows 7 or above
* Google Chrome

1. **Objective:**

* To understand the concept of Trees.

A tree is a collection of entities called nodes. Nodes are connected by edges. Each node contains a value or data, and it may or may not have a child node. The first node of the tree is called the root.

1. **Code:**

class Solution {

 public:

  bool isSymmetric(TreeNode\* root) {

    return isSymmetric(root, root);

  }

 private:

  bool isSymmetric(TreeNode\* p, TreeNode\* q) {

    if (!p || !q)

      return p == q;

    return p->val == q->val &&                //

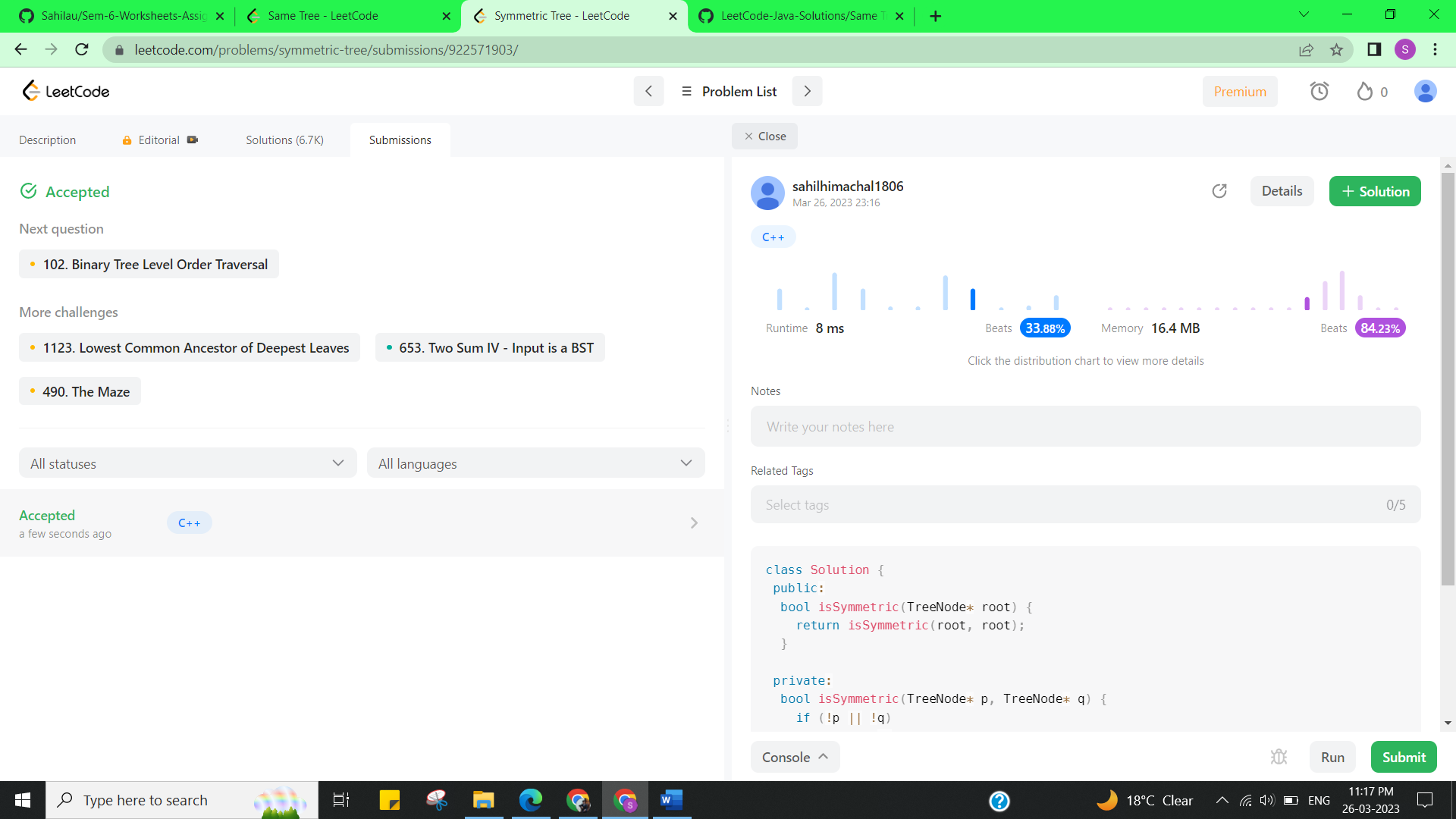
           isSymmetric(p->left, q->right) &&  //

           isSymmetric(p->right, q->left);

  }

};

1. **Result/Output/Writing Summary:**



**Learning outcomes (What I have learnt):**

* Learned the concept of trees.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |