DSA- Assignment-5

Questions List :-

1. Question was- https://leetcode.com/problems/number-of-students-unable-to-eat-lunch/description/

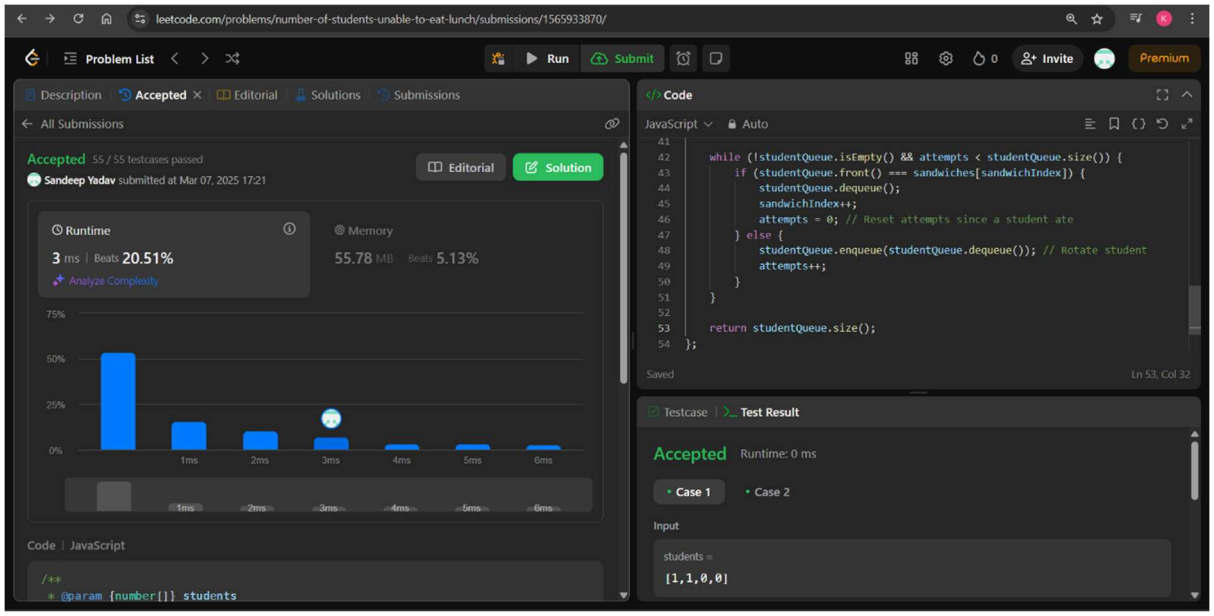
Solution Link - https://leetcode.com/problems/number-of-students-unable-to-eat-lunch/submissions/1565933870/

Description:

(i) Function to count the number of students unable to eat lunch.
(ii) Each student prefers either circular (0) or square (1) sandwiches.
(iii) Students take sandwiches in the order they appear if their preference matches.
(iv) If a student at the front doesn't take a sandwich, they go to the end of the queue.
(v) If all remaining students can't take any sandwiches, the loop stops.

Time Complexity: O(N) - Each student is processed at most twice.

Space Complexity: O(N) - use a queue to store students.



2. Question was- https://leetcode.com/problems/minimum-depth-of-binary-tree/description/

Solution Link- https://leetcode.com/problems/minimum-depth-of-binary-tree/submissions/1566978304/

Description:

(i) If the root is null, return 0.

(ii) Use a queue to perform level-order traversal (BFS).Define condition using while loop.

(iii) Start with the root node at depth 1 and add it to the queue.

(iv) While the queue is not empty:

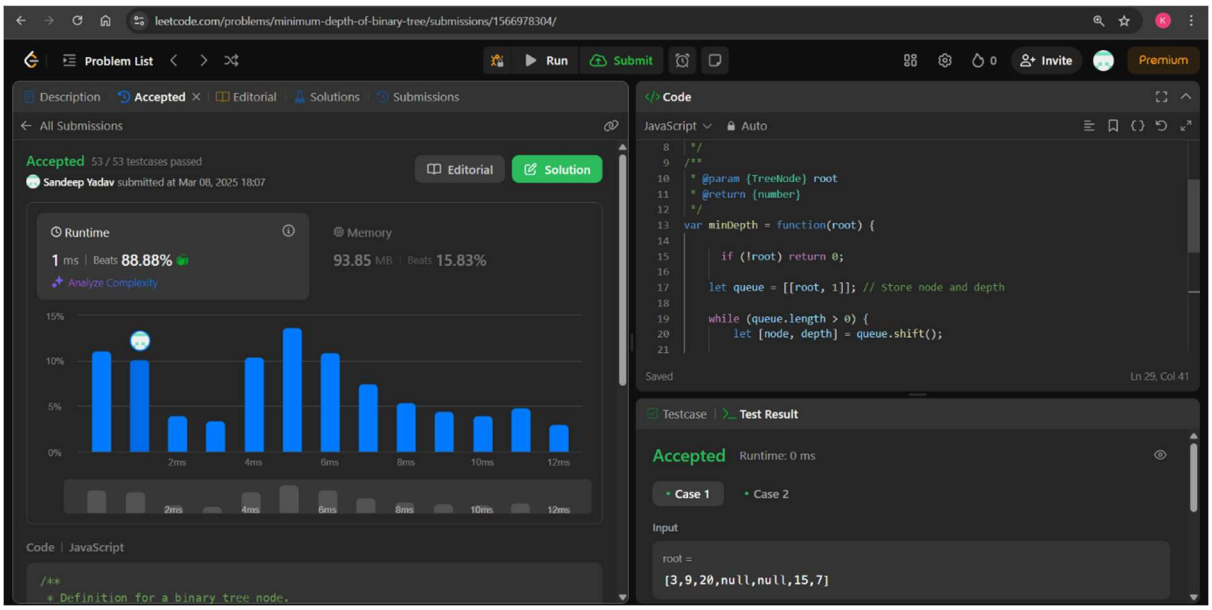
  a) Remove a node and check if it's a leaf node (no left and right children).If it is, return the current depth.

  b) Otherwise, add its children to the queue with depth + 1.

(v) The first leaf encountered gives the minimum depth.

Time Complexity: O(N), where N is the length of nums2 and M is the length of nums1

Space Complexity: O(N) extra space for HashMap and Stack



3. Question was- https://leetcode.com/problems/binary-tree-postorder-traversal/description/

Solution Link- https://leetcode.com/problems/binary-tree-postorder-traversal/submissions/1566982587/

Description:

(i) Initialize an empty stack and an output array.

(ii) Push the root onto the stack.

(iii) While the stack is not empty:

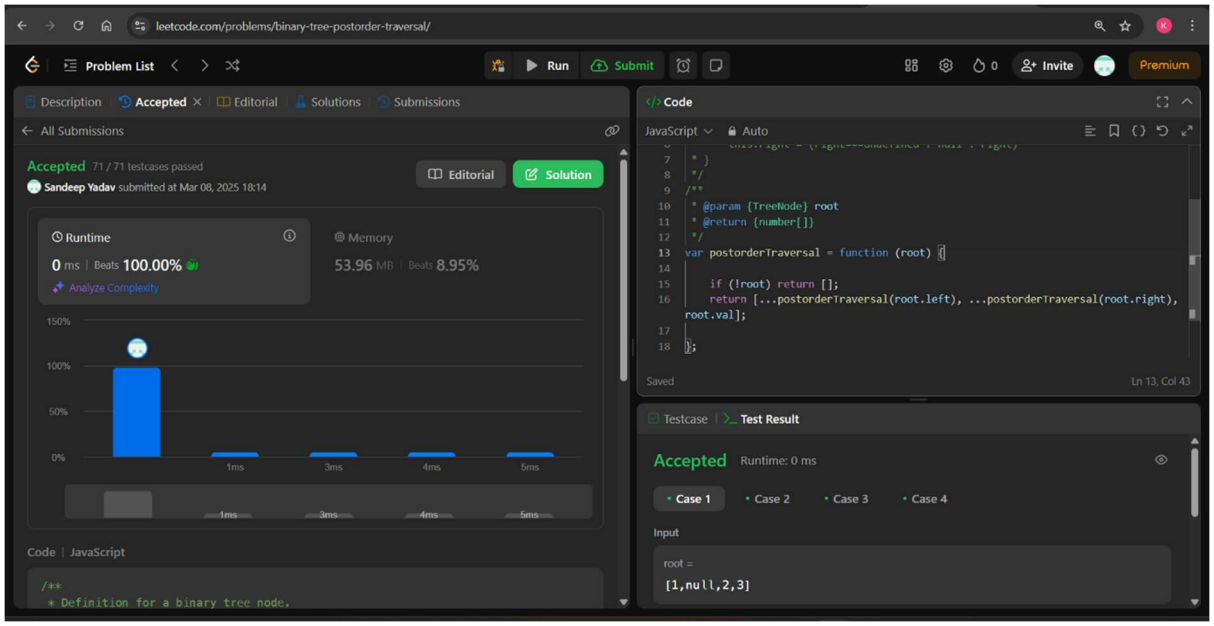
a) Pop a node, insert it at the beginning of the output array.

b) Push its left child to the stack if it exists.

c) Push its right child to the stack if it exists.

Time Complexity: O(N) as each node is processed once.

Space Complexity: O(N) or storing nodes in the stack in the worst case.



4. Question was- https://leetcode.com/problems/binary-tree-preorder-traversal/description/

Solution Link– https://leetcode.com/problems/binary-tree-preorder-traversal/submissions/1567069512/

Description:

i) Initialize an empty stack and an output array.

ii) Push the root onto the stack.

iii) While the stack is not empty:

a) Pop a node, add its value to the output.

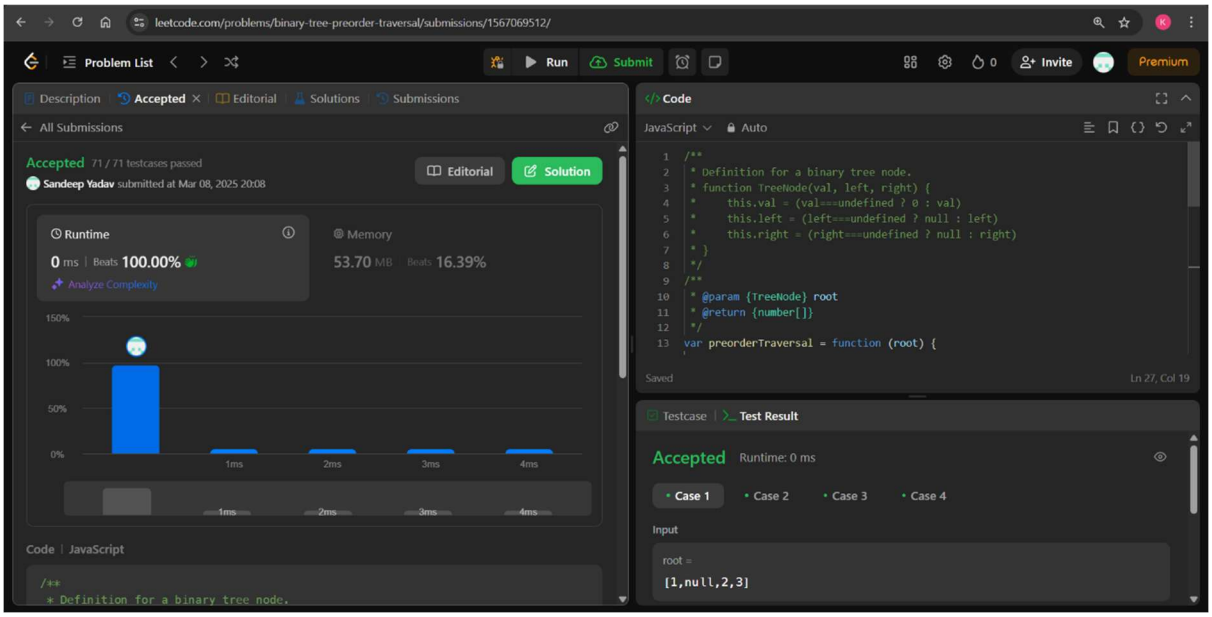
b) Push its right child to the stack **if** it exists.

c) Push its left child to the stack **if** it exists.

iV) Return the output array.

Time Complexity: O(N) *as each node is processed once.*

Space Complexity: O(N) *for storing nodes in the stack in the worst case.*



5. Question was- https://leetcode.com/problems/binary-tree-inorder-traversal/description/

Solution Link - https://leetcode.com/problems/binary-tree-inorder-traversal/submissions/1567092195/

Description:

(i) Initialize an empty stack and an empty output array.

(ii) Start from the root and push all left nodes onto the stack.

(iii) While the stack is not empty:

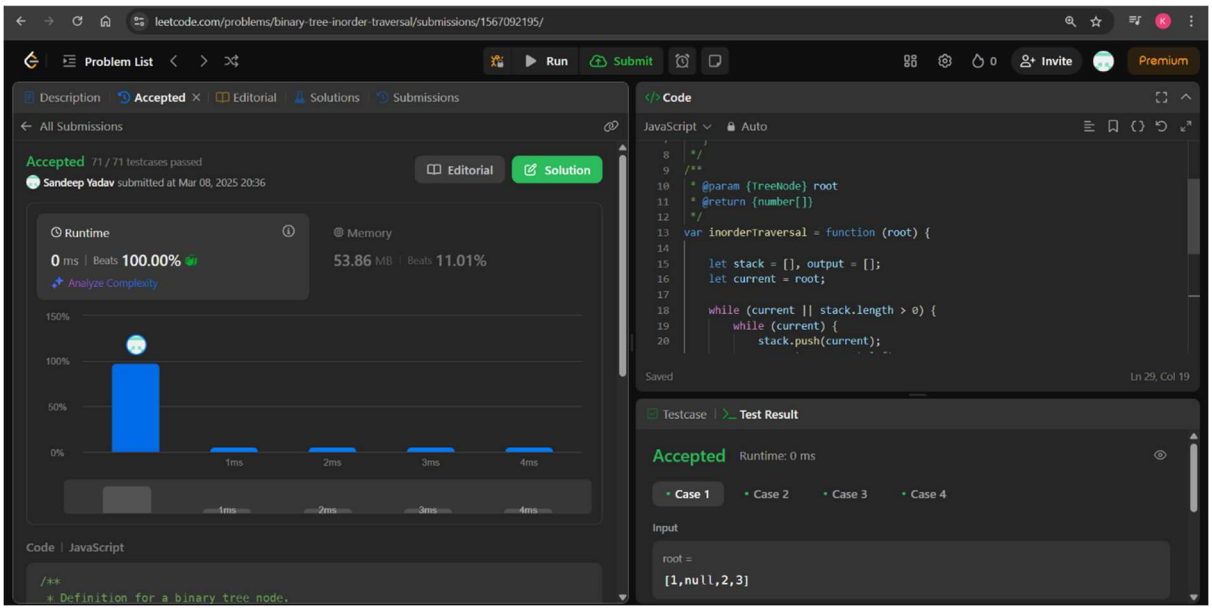
a) Pop a node from the stack and add its value to the output.

b) Move to the right subtree and push all left nodes onto the stack.

(iv) Return the output array.

Time Complexity: O(N) as each node is processed once..

Space Complexity: O(N) for storing nodes in the stack in the worst case.



Key Features :-

- Solutions are less time taking.

Difficulties :-

- Facing Difficulties in Code Optimization.
- Try to make code efficient and use better approach to solve them.

GitHub Repository Link : -

https://github.com/SandyBhai03/Internshala-Assignments/blob/main/Assignment-Course5/DSA-2/Assignment-5/app.js