Question No. 1 :-
https://leetcode.com/problems/two-sum/description/
Solution Link :-
https://leetcode.com/problems/two-sum/submissions/1378860412/
Description :-

Time Complexity :- O(n)

The code iterates through the nums array exactly once.

Checking if a key exists in the map (map.has(number)) and retrieving a value from the map (map.get(number)) both happen in constant time, O(1).

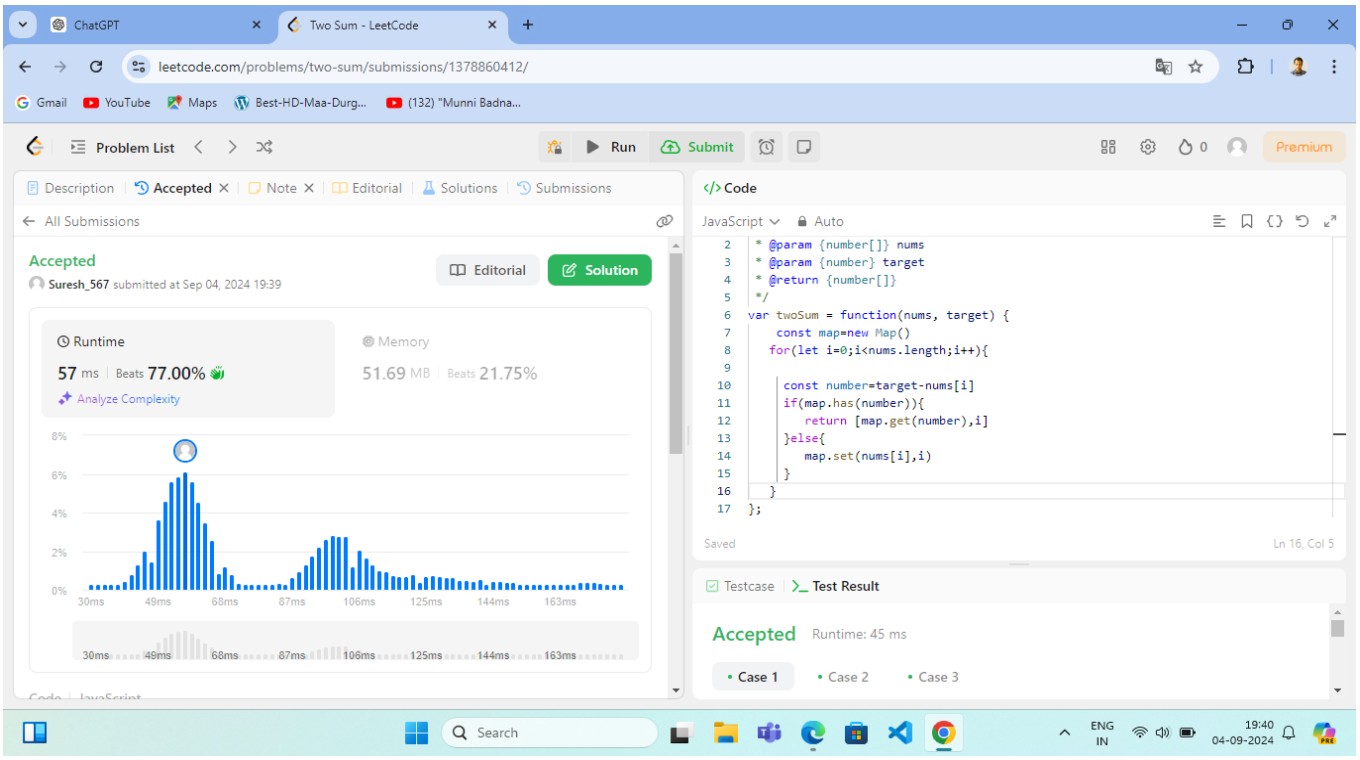
So, the overall time complexity is **O(n)**, where n is the length of the nums array.

Space Complexity :- O(n)

In the worst case, if no two elements sum to the target, all n elements would be stored in the map.

So, the space complexity is **O(n)**.

Screenshot :-



Question No. 15 :- https://leetcode.com/problems/3sum/description/

Solution Link :- https://leetcode.com/problems/3sum/submissions/1378898456/

Description :-

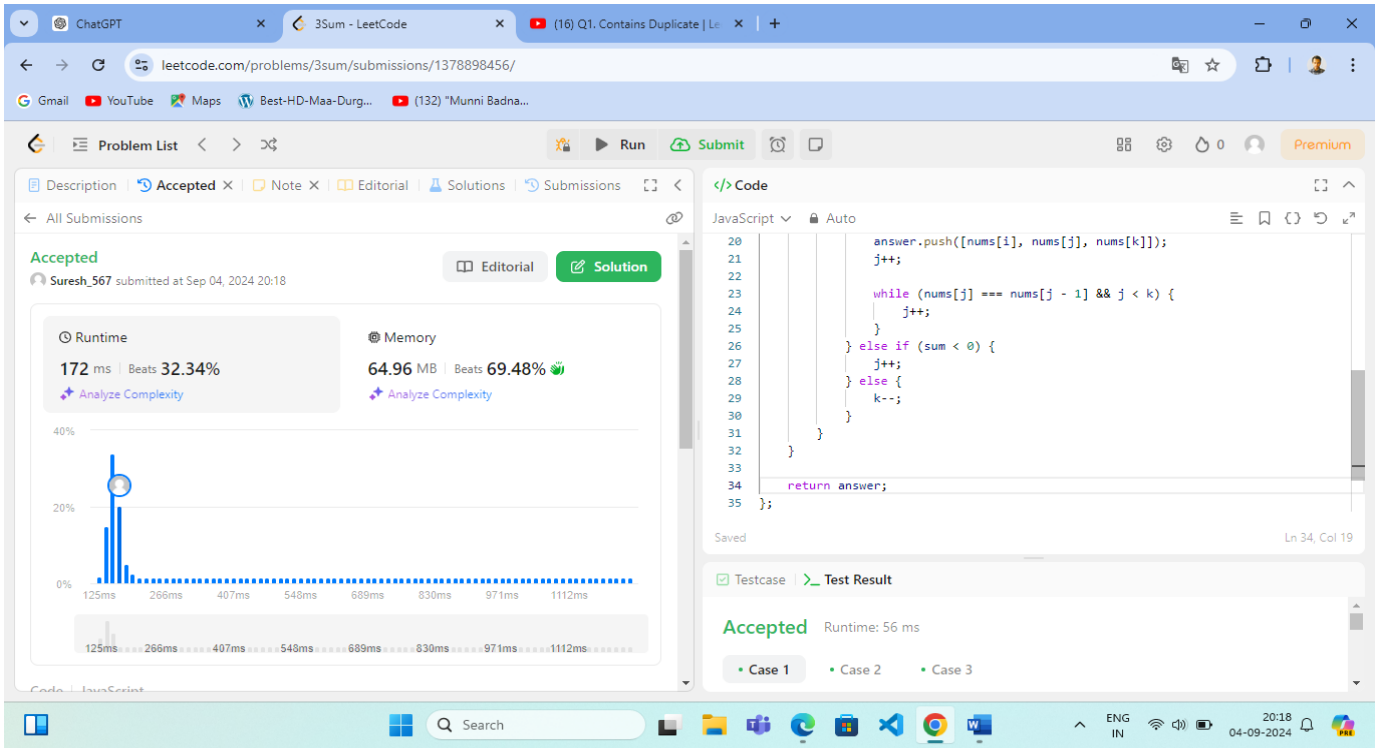
Time Complexity :- O(n^2)

Combining the sorting and the nested loops, the total time complexity is **O(n^2)**

Space Complexity :- O(n^2)

The overall space complexity is **O(n^2)**, which is due to the space required to store the result triplets in the answer array.

Screenshot :-



Question No. 925 :-

https://leetcode.com/problems/long-pressed-name/description/

Solution Link :-

https://leetcode.com/problems/long-pressed-name/submissions/1378915248/

Description :-

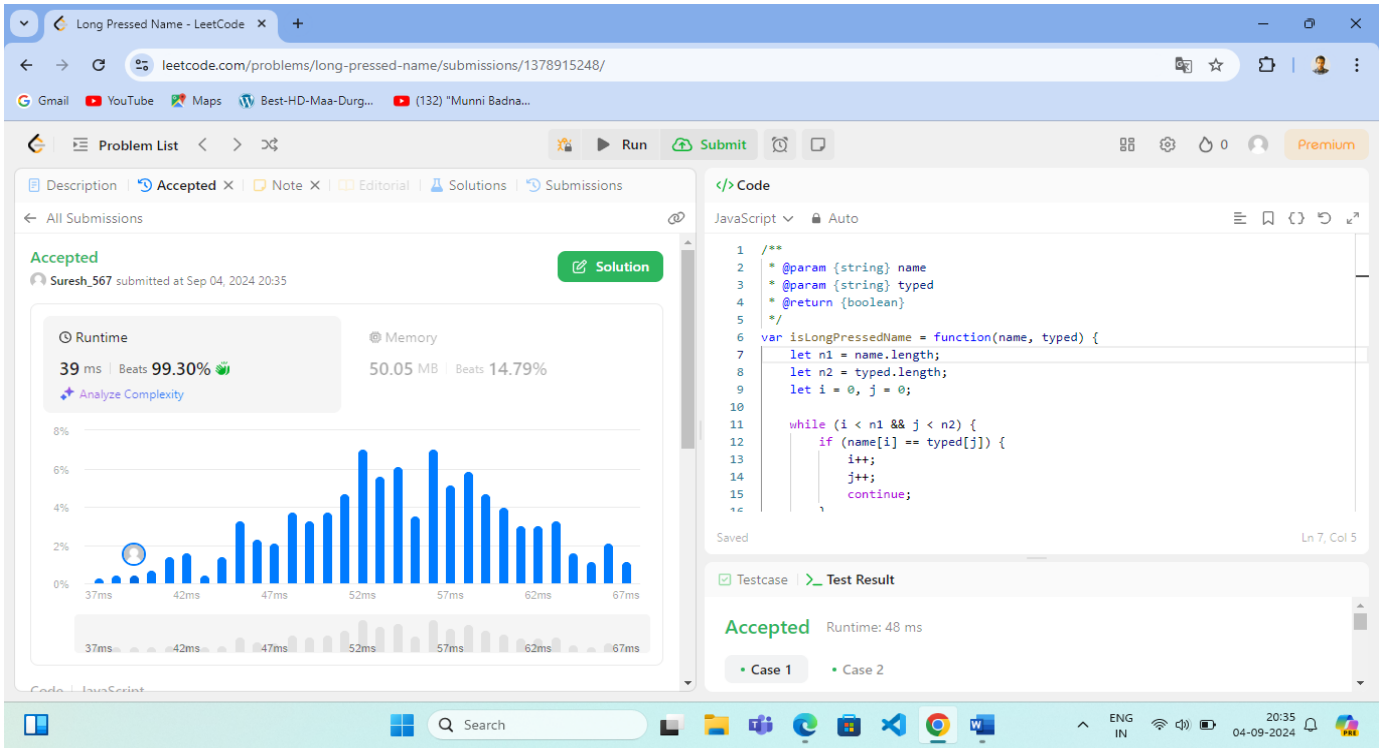
Time Complexity :- O(n1 + n2)

The time complexity is **O(n1 + n2)**, where n1 is the length of the name string, and n2 is the length of the typed string.

Space Complexity :- O(1)

The space complexity is **O(1)** because the only additional space used is for the variables i, j, n1, and n2.

Screenshot :-



Question No. 769 :-

https://leetcode.com/problems/max-chunks-to-make-sorted/description/

Solution Link :- https://leetcode.com/problems/max-chunks-to-make-sorted/submissions/1378931310/

Description :-

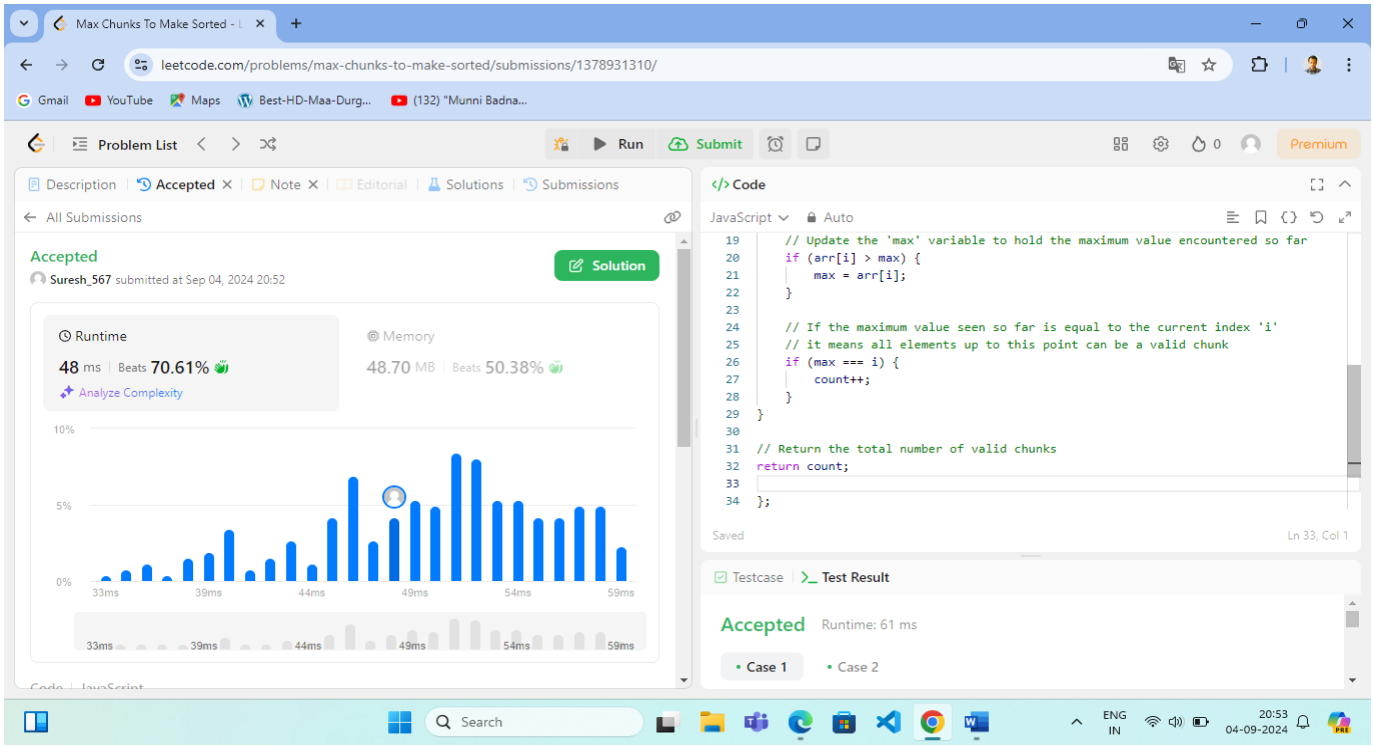
Time Complexity :- O(n)

The overall time complexity is **O(n)**, where n is the length of the array arr.

Space Complexity :- O(1)

The space complexity is **O(1)** because only a constant amount of extra space is used for variables like count, max, and i.

Screenshot :-



Question No. 75:-
https://leetcode.com/problems/sort-colors/description/
Solution Link :-
https://leetcode.com/problems/sort-colors/submissions/1378975831/

Description :-

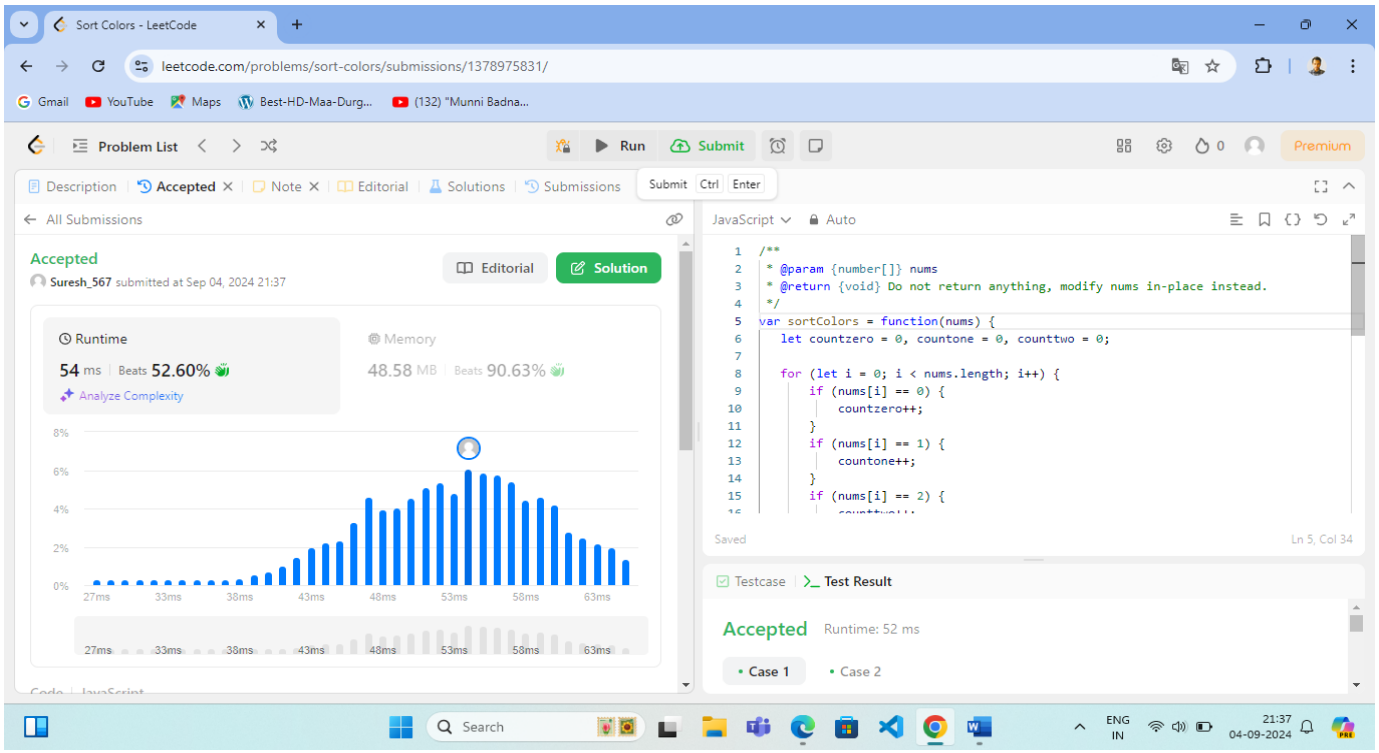
Time Complexity :- O(n)

For Counting time complexity O(n) and for replacement time complexity O(n).So, the total time complexity is **O(n)**.

Space Complexity :- O(1)

The space complexity is **O(1)** because only a constant amount of extra space is used for variables.

Screenshot :-



Question No. 53:-

https://leetcode.com/problems/maximum-subarray/

Solution Link :-

https://leetcode.com/problems/maximum-subarray/submissions/1378996672/

Description :-

Time Complexity :- O(n)

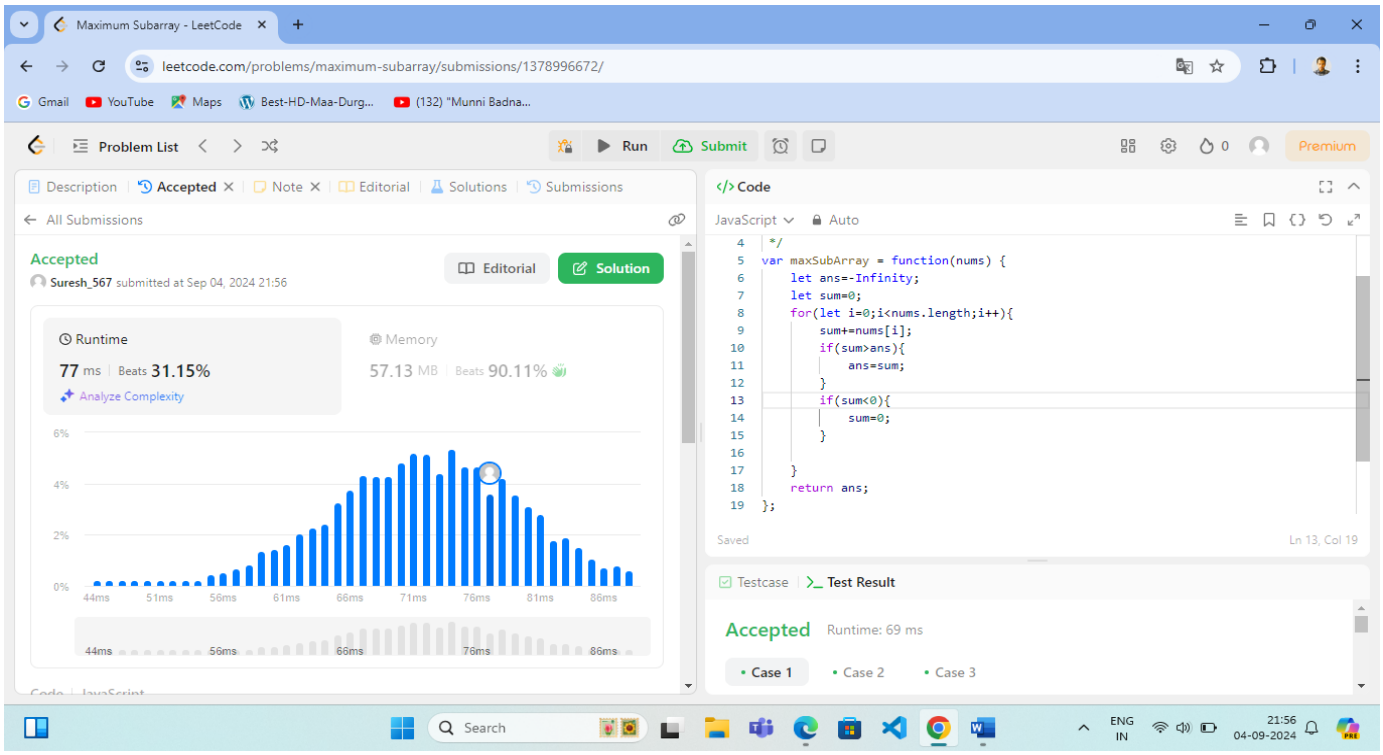
The code iterates through the nums array once using a for loop, and each operation inside the loop (addition, comparison, assignment) is O(1).

Therefore, the total time complexity is **O(n)**, where n is the length of the nums array.

Space Complexity :- O(1)

The space complexity is **O(1)** because only a constant amount of extra space is used for variables.

Screenshot :-



Question No. 238:-

https://leetcode.com/problems/product-of-array-except-self/description/

Solution Link :-

https://leetcode.com/problems/product-of-array-except-self/submissions/1379013353/

Description :-

Time Complexity :- O(n)

Total time complexity O(n), because all operations involve a single pass through the array.

Space Complexity :- O(n)

The space complexity O(n), because additional space is used for the prefix product array, suffix product array, and the result array.

Screenshot :-