DSA- Assignment-2

Questions List :-

1. Question was- https://leetcode.com/problems/two-sum/description/

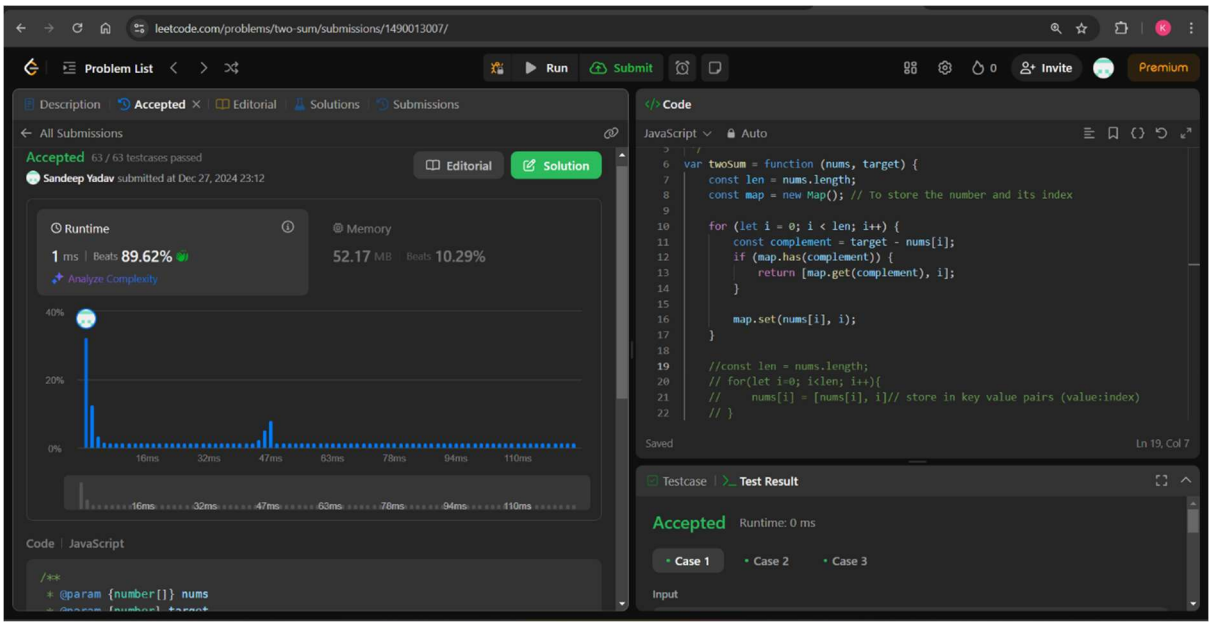
Solution Link – https://leetcode.com/problems/two-sum/submissions/1490013007/

Description:

(i) Initialize a Hash Map (map).
(ii) Iterate Through the Array.
(iii) Return Result If no two numbers add up to the target, return an empty array.

Time Complexity: O(n)

Space Complexity: O(n)



2. Question was- https://leetcode.com/problems/3sum/description/

Solution Link-https://leetcode.com/problems/3sum/submissions/1490339063/

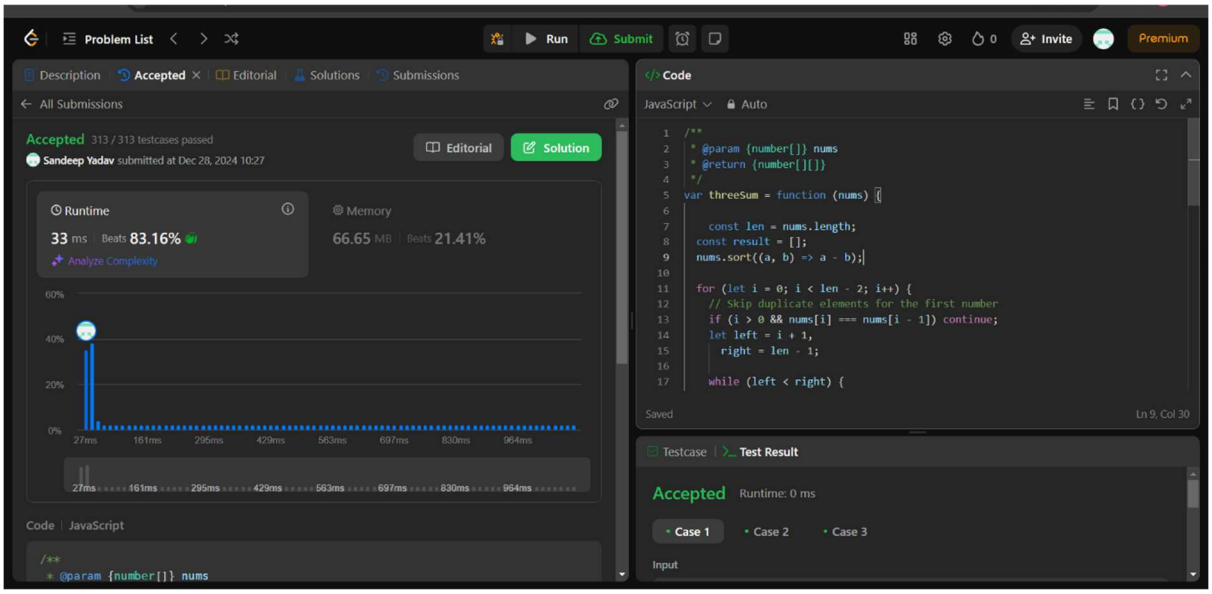
Solution Link – https://leetcode.com/problems/two-sum/submissions/1490013007/

Description:

(i) Sort the array & initialize the left & right variables for iterate.
(ii) Run while loop till left<right and check sum === 0 or not.
(iii) Return result array.

Time Complexity: O(n2)

Space Complexity: O(n) use n size of array to store answer



3. Question was- *https://leetcode.com/problems/long-pressed-name/description/*

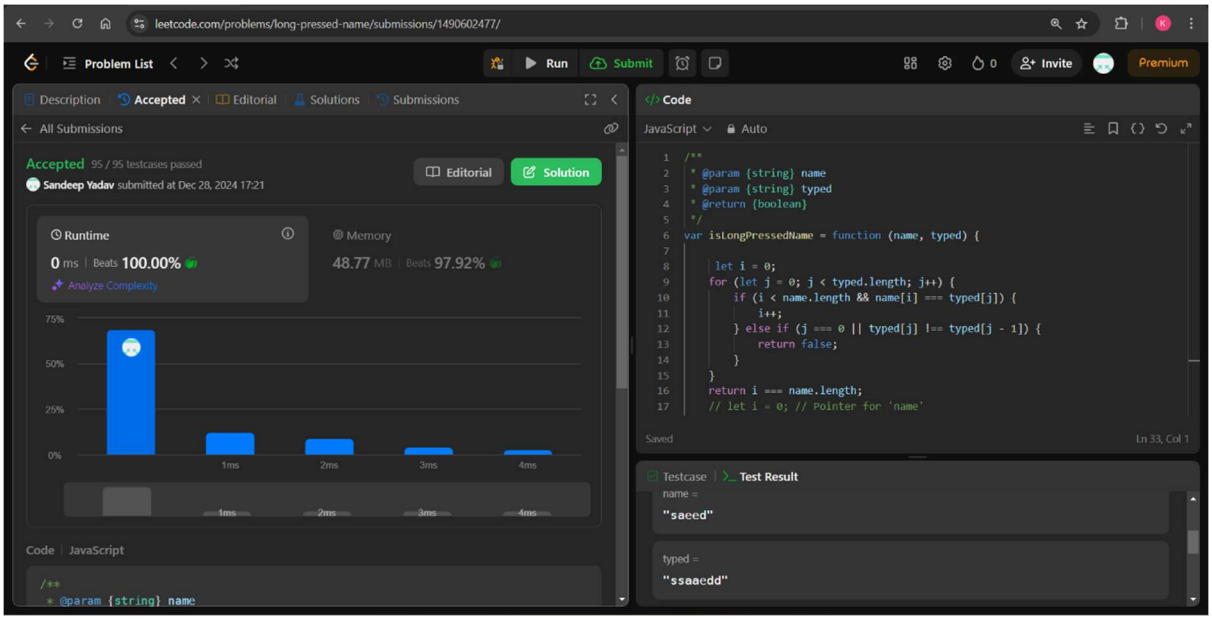
Solution Link– https://leetcode.com/problems/long-pressed-name/submissions/1490602477/

Description:

(i) Run for loop till the length of typed variable.
(ii) Check typed[j] !== typed[j - 1] return false.
(iii) Return i === name.length.

Time Complexity: O(n+m) n is length of name & m is length of typed

Space Complexity: *O(1)* no use extra space



4. Question was- *https://leetcode.com/problems/max-chunks-to-make-sorted/description/*

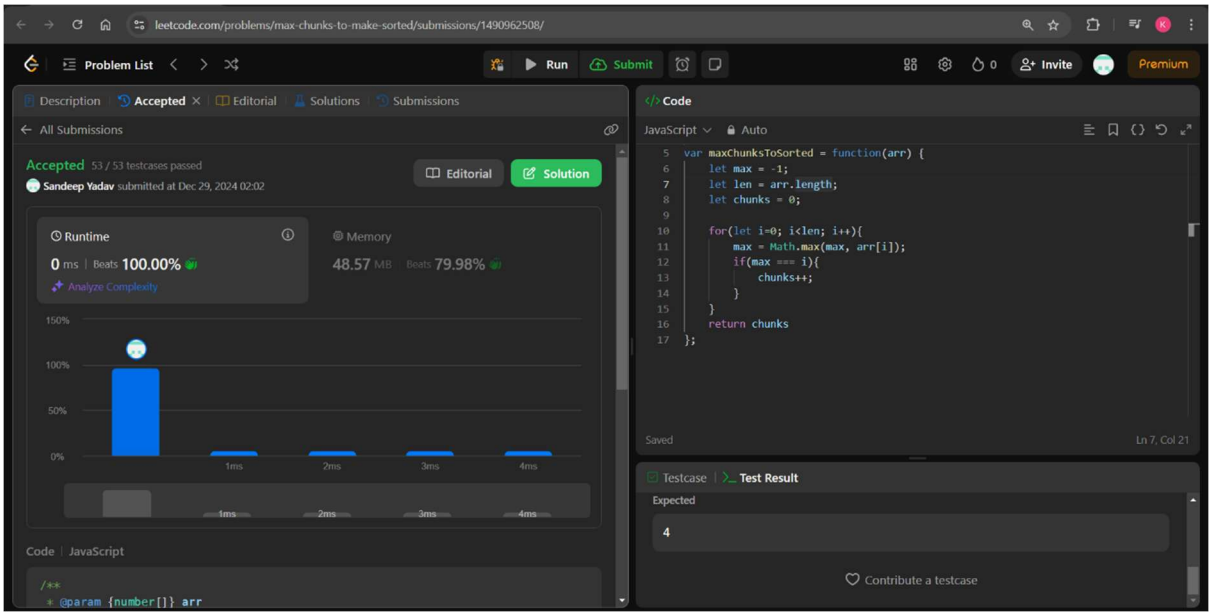
Solution Link– https://leetcode.com/problems/max-chunks-to-make-sorted/submissions/1490962508/

Description:

(i) Define max = -1 & chunks = 0.
(ii) Iterate all the elements.
(iii) Find max from (max & arr[i])
(iv) Check if (max===i) then add 1 in chunks.

Time Complexity: O(n) use linear TC

Space Complexity: O(1) no extra space



5. Question was- https://leetcode.com/problems/sort-colors/description/

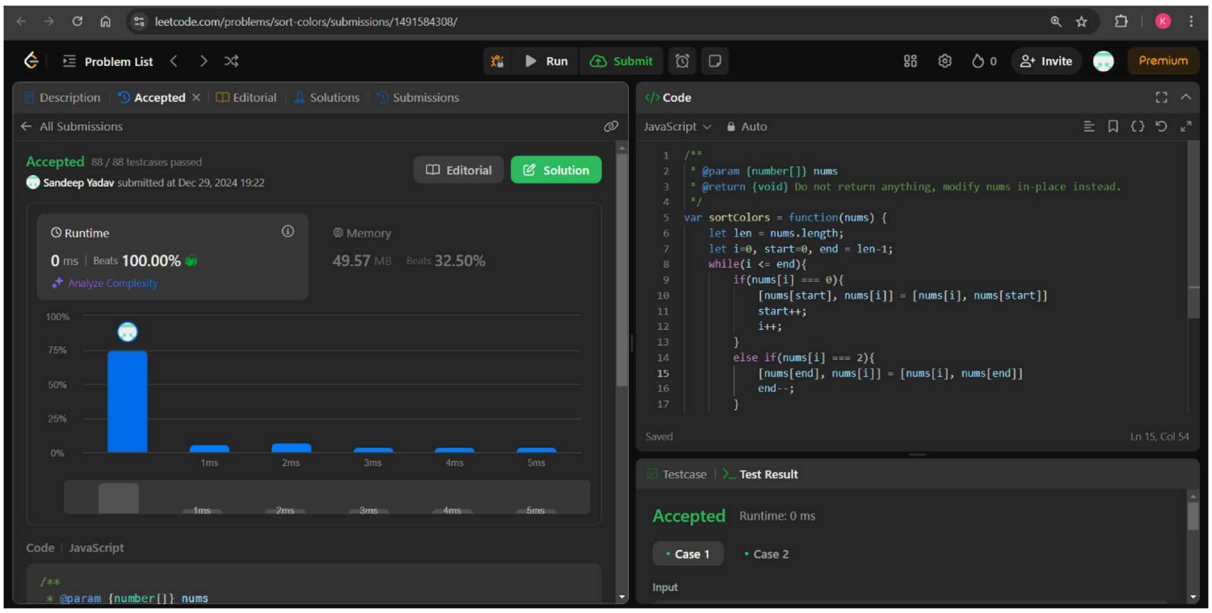
Solution Link- https://leetcode.com/problems/sort-colors/submissions/1491584308/

Description:

(i) Run while loop till then (i<=end).
(ii) Check if (nums[i]===0) swap nums[start] from nums[i] & start++ & i++.
(iii) If (nums[i] === 2) swap nums[i] from nums[end] & end--.
(iv) If(nums[i] ===1) i++.

Time Complexity: O(n)

Space Complexity: O(1) no use extra space



6. Question was- https://leetcode.com/problems/maximum-subarray/description

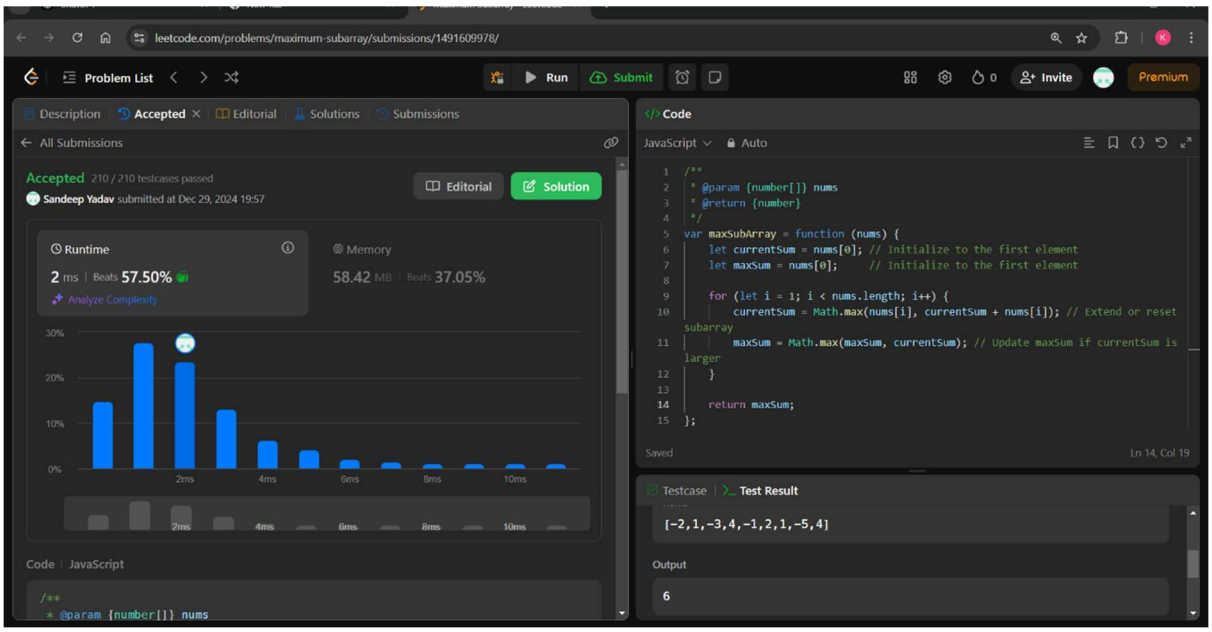
Solution Link- https://leetcode.com/problems/maximum-subarray/submissions/1491609978/

Description:

(i) Initialize currentSum & maxSum from nums[0].
(ii) Iterate all the elemnts and add currentSum[i] + nums[i].
(iii) Find max from currentSum[i] and maxSum and update.

Time Complexity: O(n)

Space Complexity: O(1) no use extra space



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

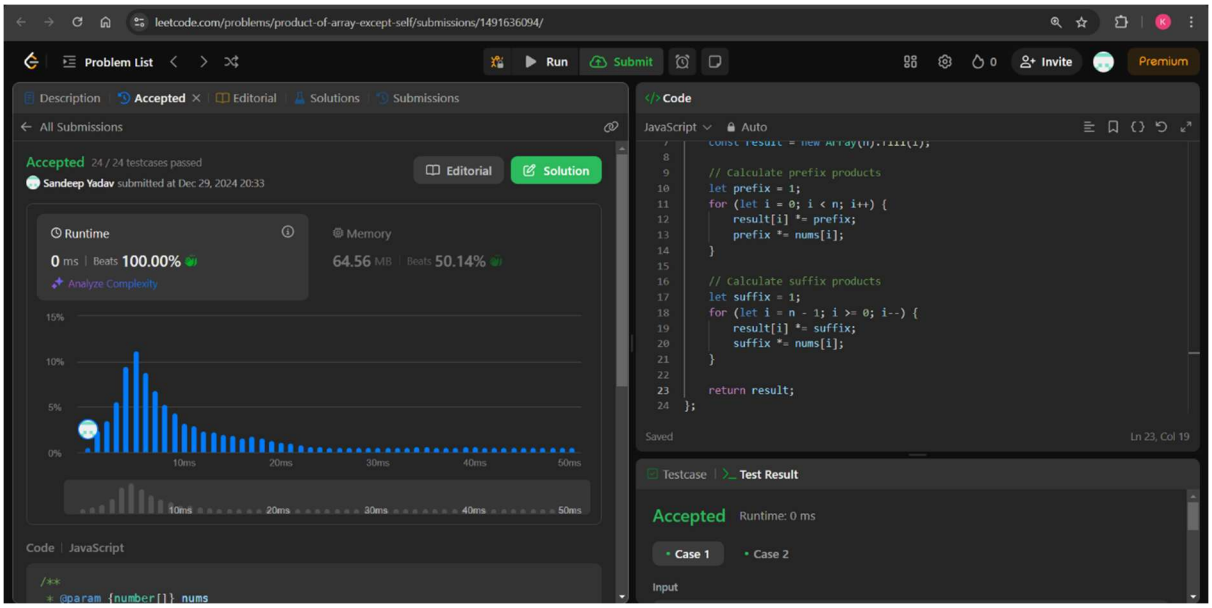
Solution Link- https://leetcode.com/problems/product-of-array-except-self/submissions/1491636094/

Description:

(i) Create new Array result size of length and fill(1).
(ii) Initialize prefix = 1 and suffix =1.
(iii) Iterate all the elements result[i] *= prefix & prefix *= nums[i] .
(iv) Iterate n-1 to 0 .
(v) Perform result[i] *= suffix & suffix *= nums[i] .

Time Complexity: O(n)

Space Complexity: O(1) no use extra space



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/tree/main/Assignment-Course5/DSA-1/Assignment-2