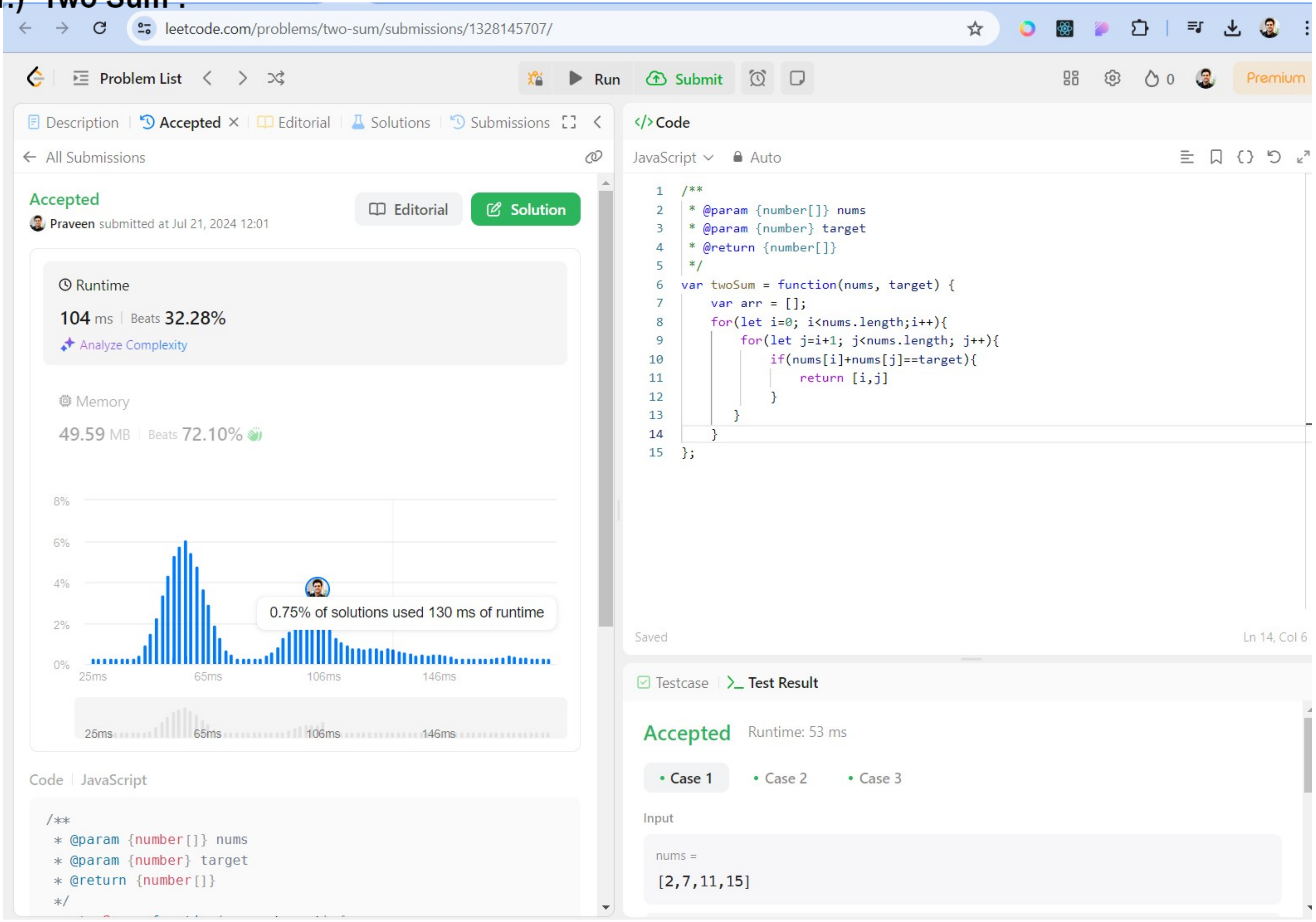
1.) Two Sum :



Accepted

Praveen submitted at Jul 21, 2024 12:01

Runtime

104 ms | Beats 32.28%

Memory

49.59 MB | Beats 72.10%

0.75% of solutions used 130 ms of runtime

```javascript
/**
 * @param {number[]} nums
 * @param {number} target
 * @return {number[]}
 */
var twoSum = function(nums, target) {
    var arr = [];
    for(let i=0; i<nums.length;i++){
        for(let j=i+1; j<nums.length; j++){
            if(nums[i]+nums[j]==target){
                return [i,j]
            }
        }
    }
};
```

Saved

Ln 14, Col 6

Accepted Runtime: 53 ms

Case 1 | Case 2 | Case 3

Input

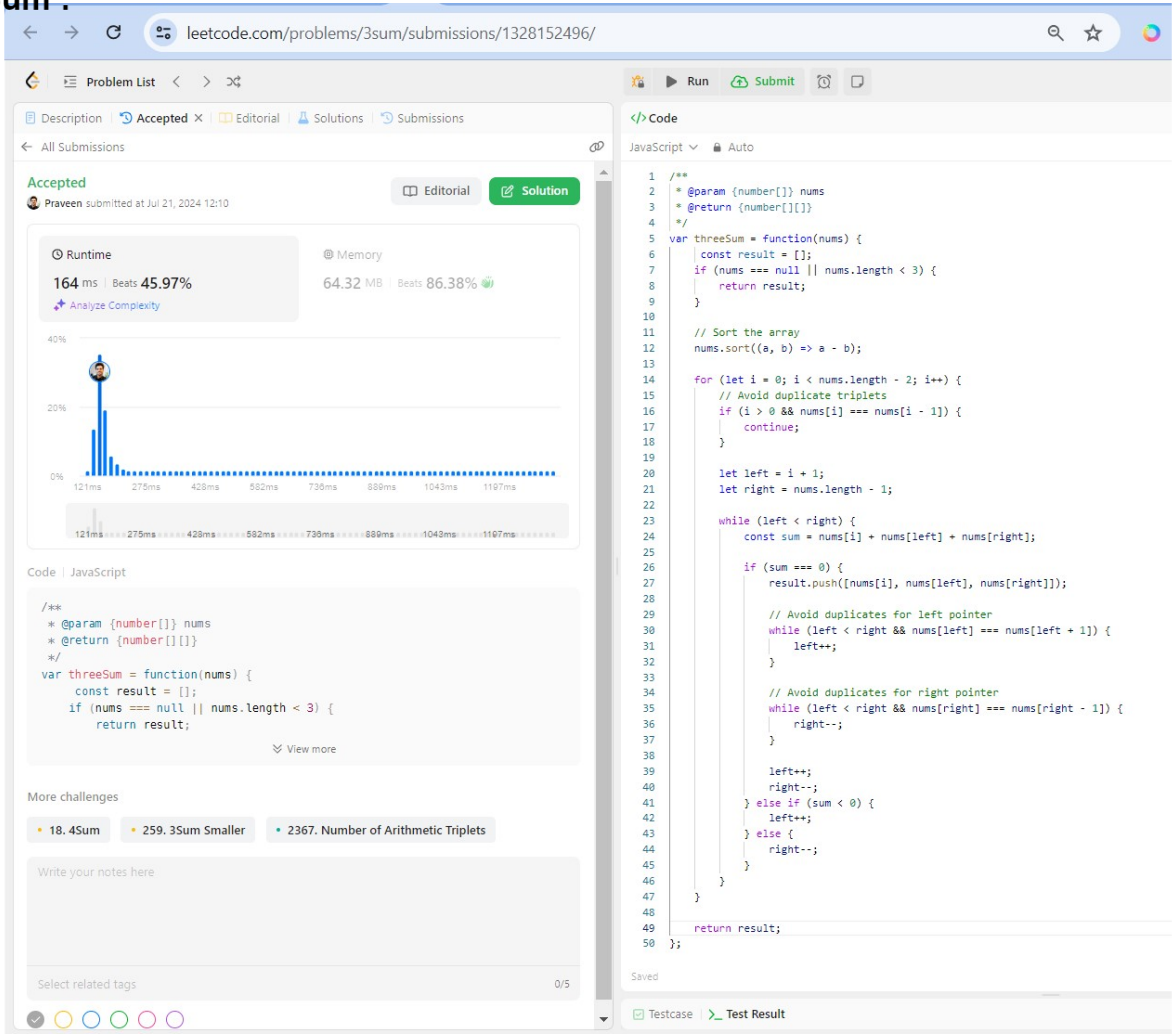
nums =

[2,7,11,15]

Code | JavaScript

```
/**
 * @param {number[]} nums
 * @param {number} target
 * @return {number[]}
 */
```

2.) 3Sum :



Accepted
Praveen submitted at Jul 21, 2024 12:10

Runtime
164 ms | Beats 45.97%

Memory
64.32 MB | Beats 86.38%

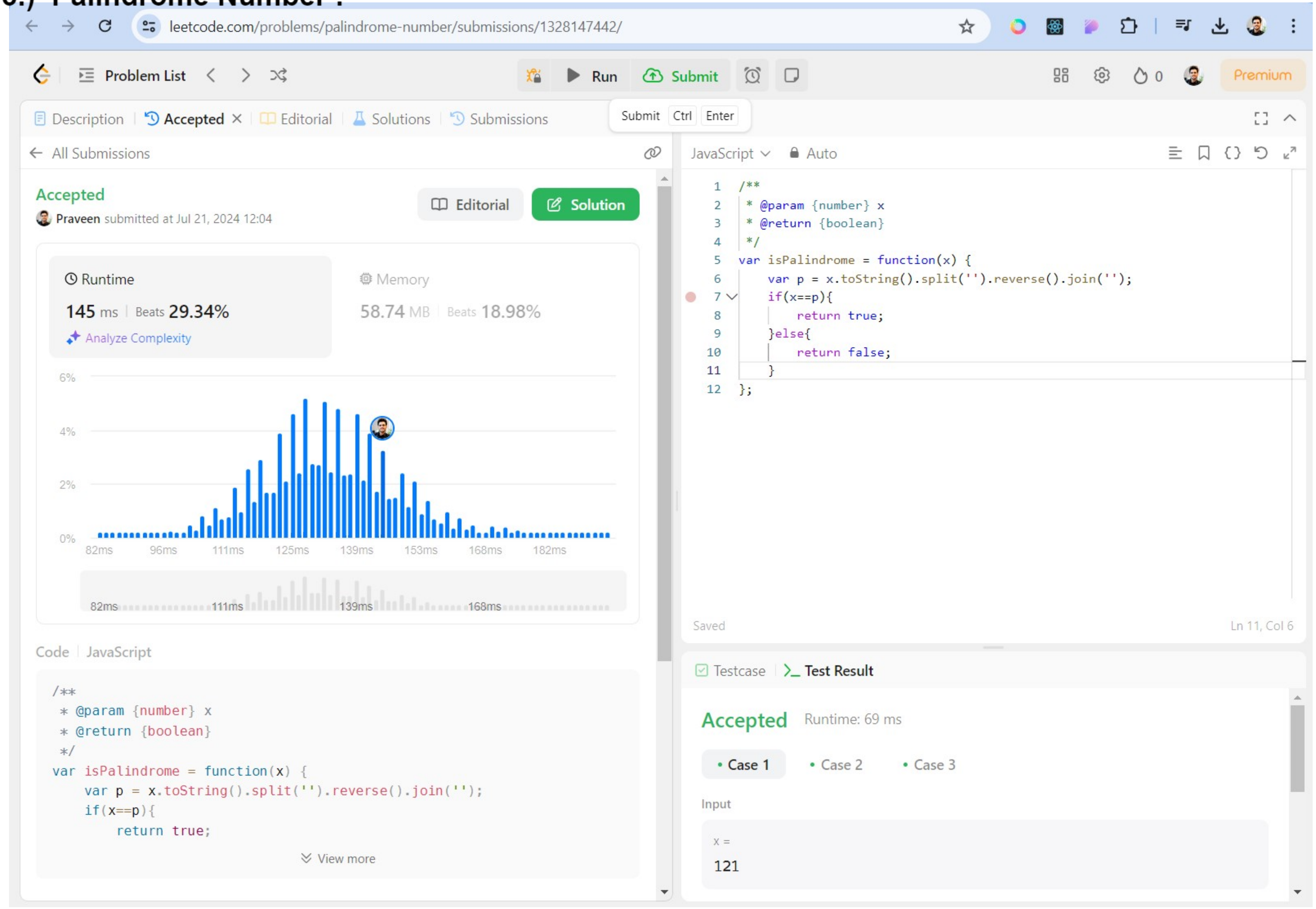
Code | JavaScript

```javascript
/**
 * @param {number[]} nums
 * @return {number[][]}
 */
var threeSum = function(nums) {
    const result = [];
   if (nums === null || nums.length < 3) {
       return result;
```

More challenges
- 18. 4Sum
- 259. 3Sum Smaller
- 2367. Number of Arithmetic Triplets

```javascript
/**
 * @param {number[]} nums
 * @return {number[][]}
 */
var threeSum = function(nums) {
     const result = [];
    if (nums === null || nums.length < 3) {
        return result;
    }

    // Sort the array
    nums.sort((a, b) => a - b);

    for (let i = 0; i < nums.length - 2; i++) {
        // Avoid duplicate triplets
        if (i > 0 && nums[i] === nums[i - 1]) {
            continue;
        }

        let left = i + 1;
        let right = nums.length - 1;

        while (left < right) {
            const sum = nums[i] + nums[left] + nums[right];

            if (sum === 0) {
                result.push([nums[i], nums[left], nums[right]]);

                // Avoid duplicates for left pointer
                while (left < right && nums[left] === nums[left + 1]) {
                    left++;
                }

                // Avoid duplicates for right pointer
                while (left < right && nums[right] === nums[right - 1]) {
                    right--;
                }

                left++;
                right--;
            } else if (sum < 0) {
                left++;
            } else {
                right--;
            }
        }
    }

    return result;
};
```

3.) Palindrome Number :



leetcode.com/problems/palindrome-number/submissions/1328147442/

Accepted

Praveen submitted at Jul 21, 2024 12:04

Runtime

145 ms | Beats 29.34%

Memory

58.74 MB | Beats 18.98%

```
/**
 * @param {number} x
 * @return {boolean}
 */
var isPalindrome = function(x) {
    var p = x.toString().split('').reverse().join('');
    if(x==p){
        return true;
    }else{
        return false;
    }
};
```

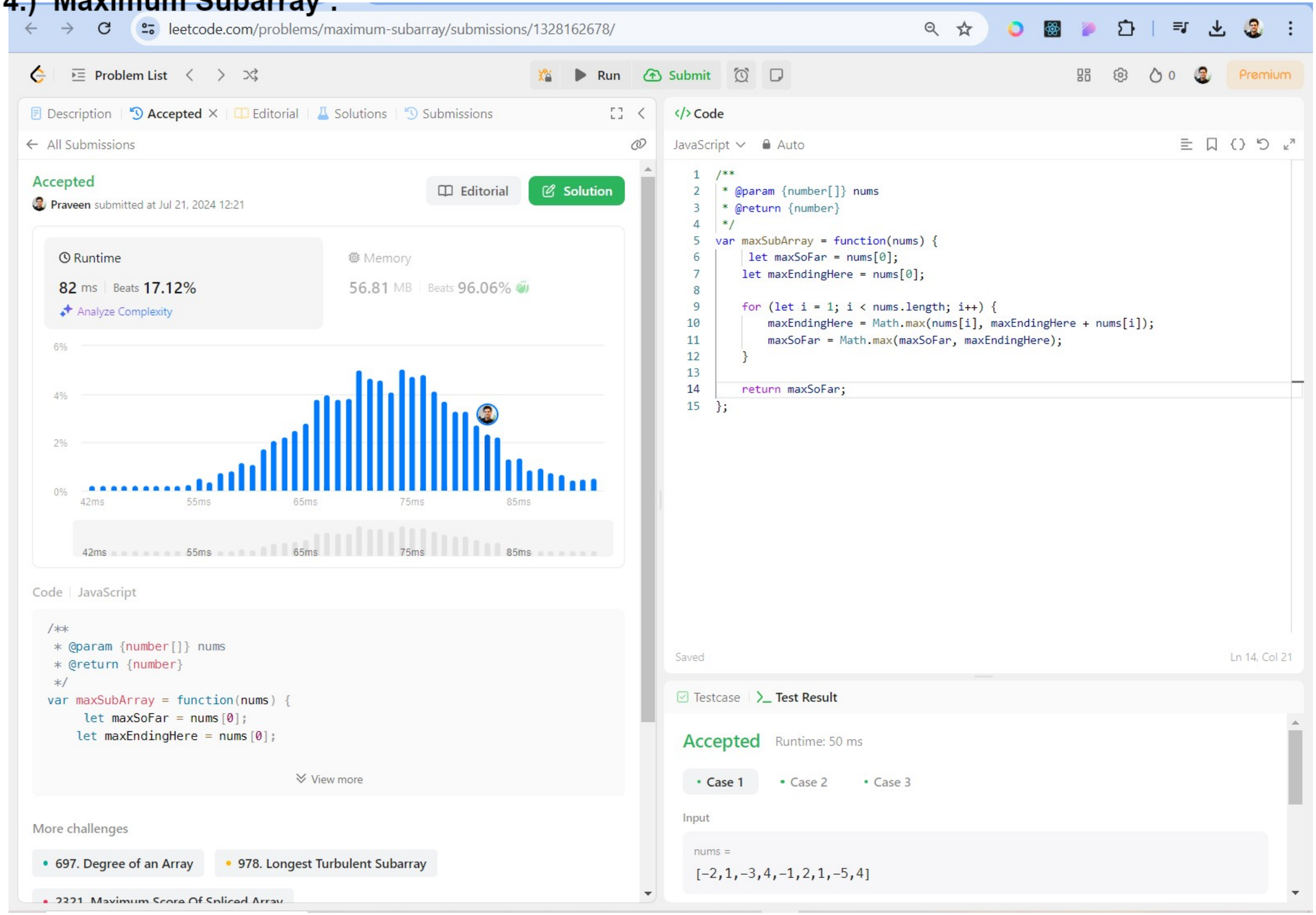
Accepted Runtime: 69 ms

Case 1 Case 2 Case 3

Input

x =
121

4.) Maximum Subarray :



leetcode.com/problems/maximum-subarray/submissions/1328162678/

Accepted
Praveen submitted at Jul 21, 2024 12:21

Runtime
82 ms | Beats 17.12%

Memory
56.81 MB | Beats 96.06%

```
/**
 * @param {number[]} nums
 * @return {number}
 */
var maxSubArray = function(nums) {
     let maxSoFar = nums[0];
    let maxEndingHere = nums[0];

    for (let i = 1; i < nums.length; i++) {
        maxEndingHere = Math.max(nums[i], maxEndingHere + nums[i]);
        maxSoFar = Math.max(maxSoFar, maxEndingHere);
    }

    return maxSoFar;
};
```

Accepted Runtime: 50 ms

Case 1 • Case 2 • Case 3

Input

nums =
[-2,1,-3,4,-1,2,1,-5,4]

More challenges
697. Degree of an Array
978. Longest Turbulent Subarray
2321. Maximum Score Of Spliced Array