Name -Yash Pratap Singh Batch- August 2024

**DSA Assignment -1 (Arrays)**

**Question 1-** There are n kids with candies. You are given an integer array candies, where each candies[i] represents the number of candies the ith kid has, and an integer extraCandies, denoting the number of extra candies that you have.

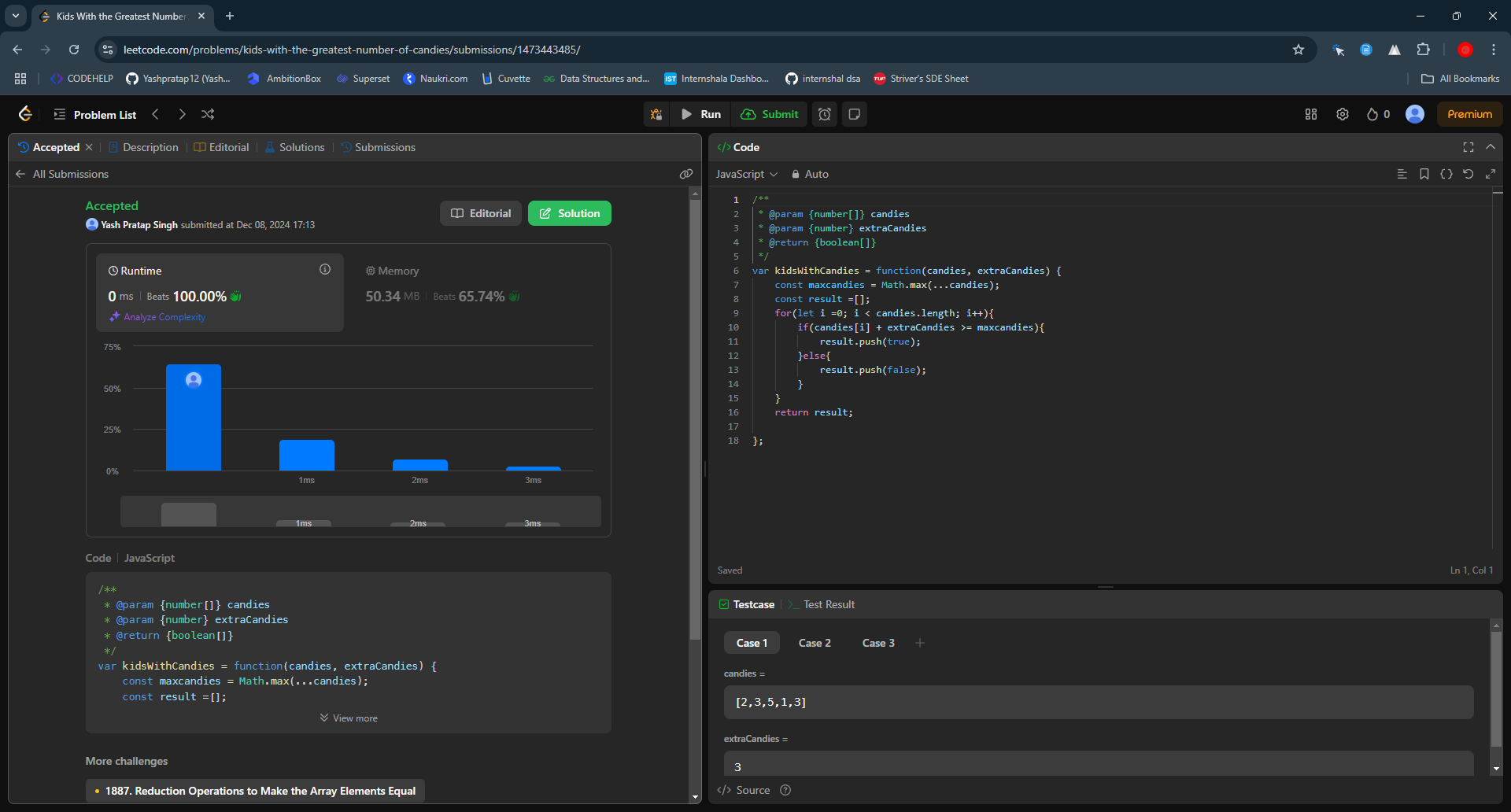
Return a boolean array result of length n, where result[i] is true if, after giving the ith kid all the extraCandies, they will have the greatest number of candies among all the kids, or false otherwise.

Leetcode 1431. https://leetcode.com/problems/kids-with-the-greatest-number-of-candies/description/

solution- <https://leetcode.com/problems/kids-with-the-greatest-number-of-candies/submissions/1473443485/>

Time complexity: O(n).

space complexity: O(n).



Description:

In the code for finding maximum candies we will iterating through the candies array O(n) since the loop runs n times and also checking condition so the overall complexity becomes O(n).

space complexity as the result array stores n boolean value requires n O(n) space meanwhile others variable takes max space O(1) which is negligible when we calculate over all space complexity so the space complexity will be O(1).

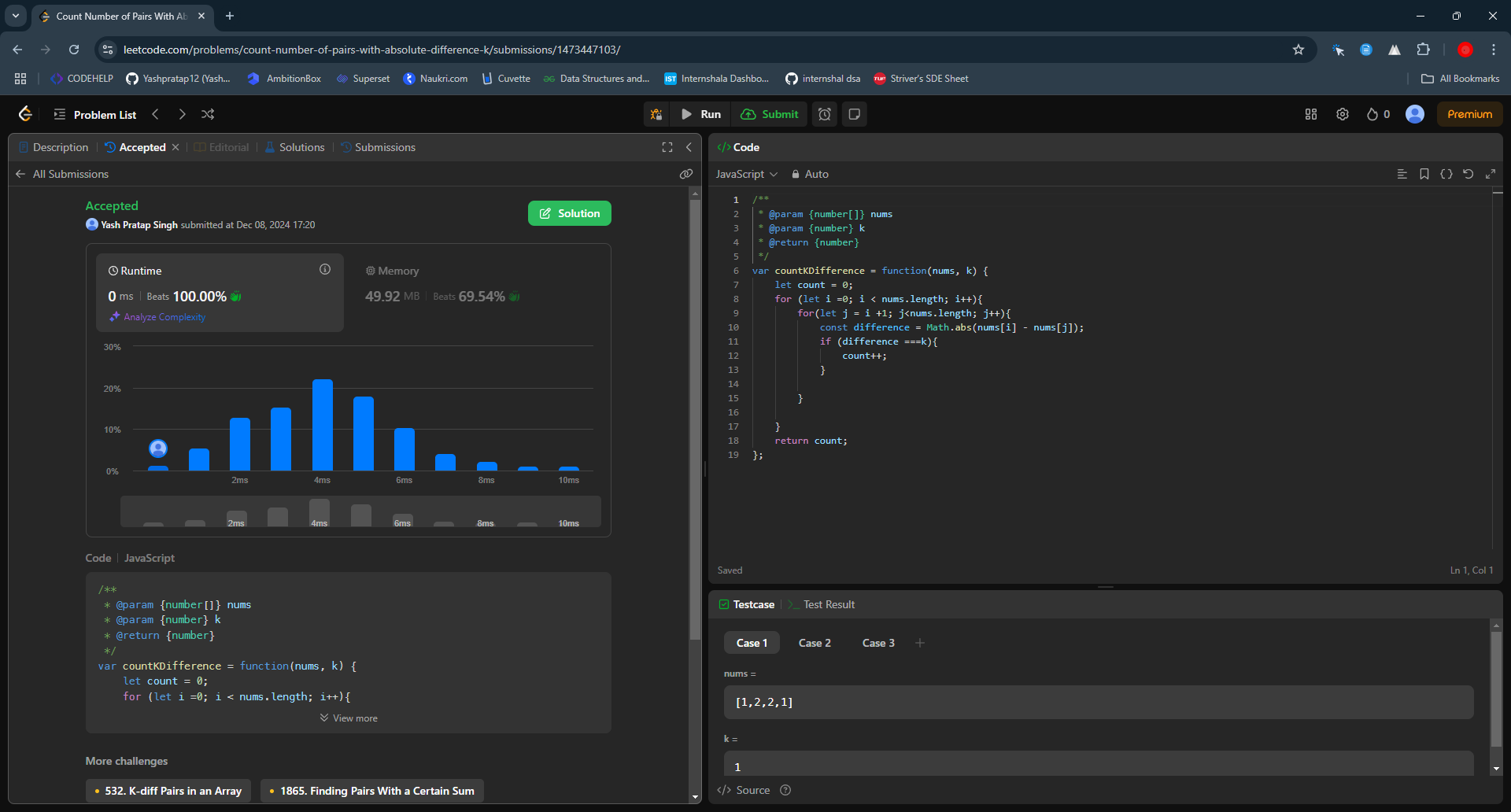
Question 2- Given an integer array nums and an integer k, return the number of pairs (i, j) where i < j such that |nums[i] - nums[j]| == k

Leetcode 2006. https://leetcode.com/problems/count-number-of-pairs-with-absolute-difference-k/description/

solution:- <https://leetcode.com/problems/count-number-of-pairs-with-absolute-difference-k/submissions/1473447103/>

Time complexity: O(n^2).

Space complexity: O(1).



Description:

Here i used 2 loops inside code so the outer loops runs n times as it iterates over n elements of the array, where as inner loop for each i inner loop run n-i-1 times which is n(n-1)/2 iteration so in total o(n)\*o(n)= o(n^2). Time complexity becomes O(n^2).

Space complexity as the code doesn’t used any additional data structure which requires extra space and the variable uses o(1) space so the Space complexity is O(1).

Question 3- You are given two 0-indexed integer arrays nums1 and nums2 of sizes n and m, respectively. Consider calculating the following values:

The number of indices i such that 0 <= i < n and nums1[i] occurs at least once in nums2.

The number of indices i such that 0 <= i < m and nums2[i] occurs at least once in nums1.

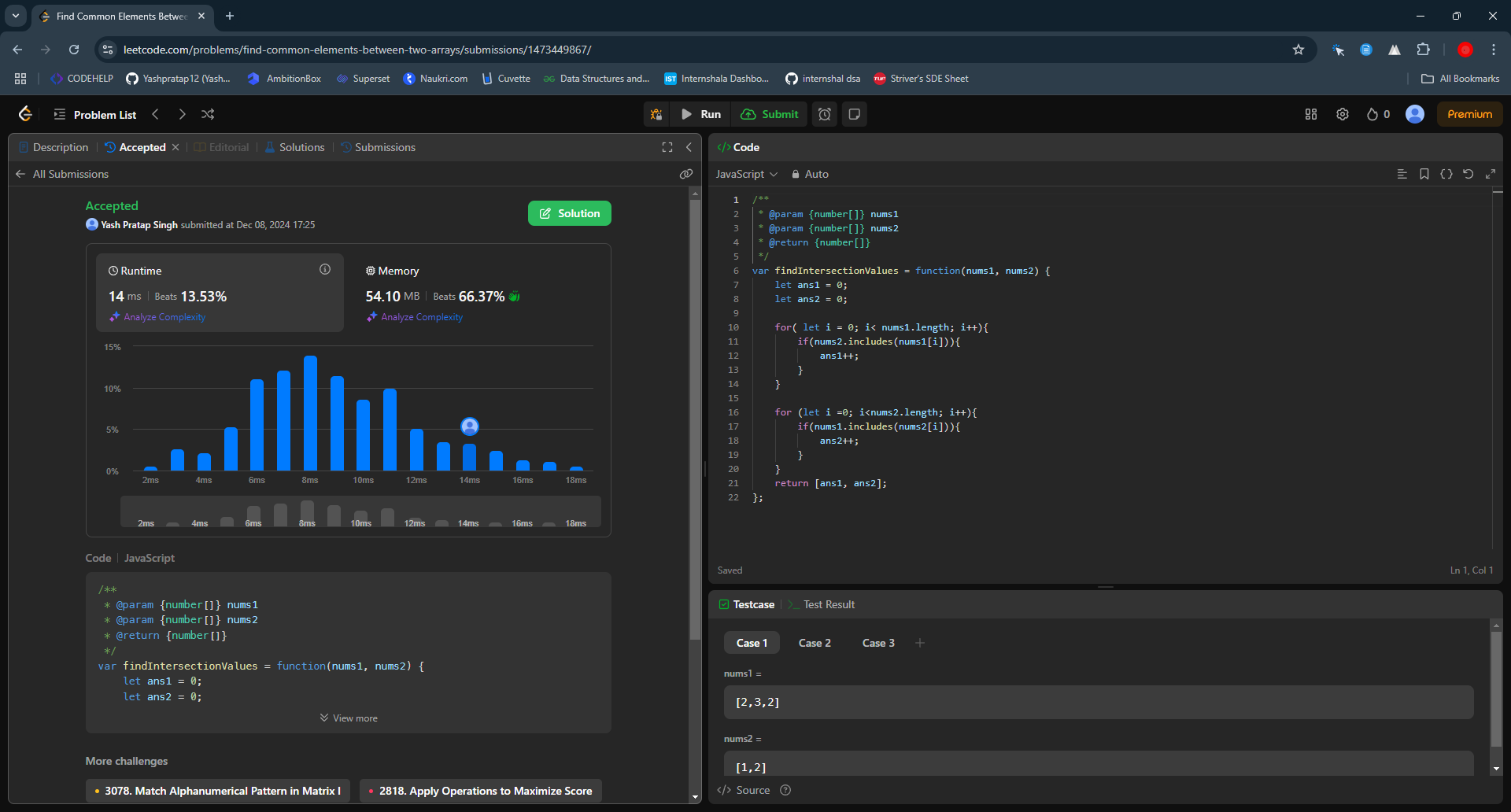
Return an integer array answer of size 2 containing the two values in the above order.

Leetcode 2956. https://leetcode.com/problems/find-common-elements-between-two-arrays/description/

solution:- <https://leetcode.com/problems/find-common-elements-between-two-arrays/submissions/1473449867/>

Time complexity: O(n.m)

Space complexity: O(1)

.

Description:

There are two loops inside code 1st loops runs n time to iterate over the n elements of an array and the second loop runs m times so the complexity becomes n\*m hence time complexity for the code becomes O(n.m).

space complexity as the no additional space is used and include does not takes extra space so the space complexity becomes O(1).

Question 4- Given an array of integers nums, return the number of good pairs. A pair (i, j) is called good if nums[i] == nums[j] and i < j.

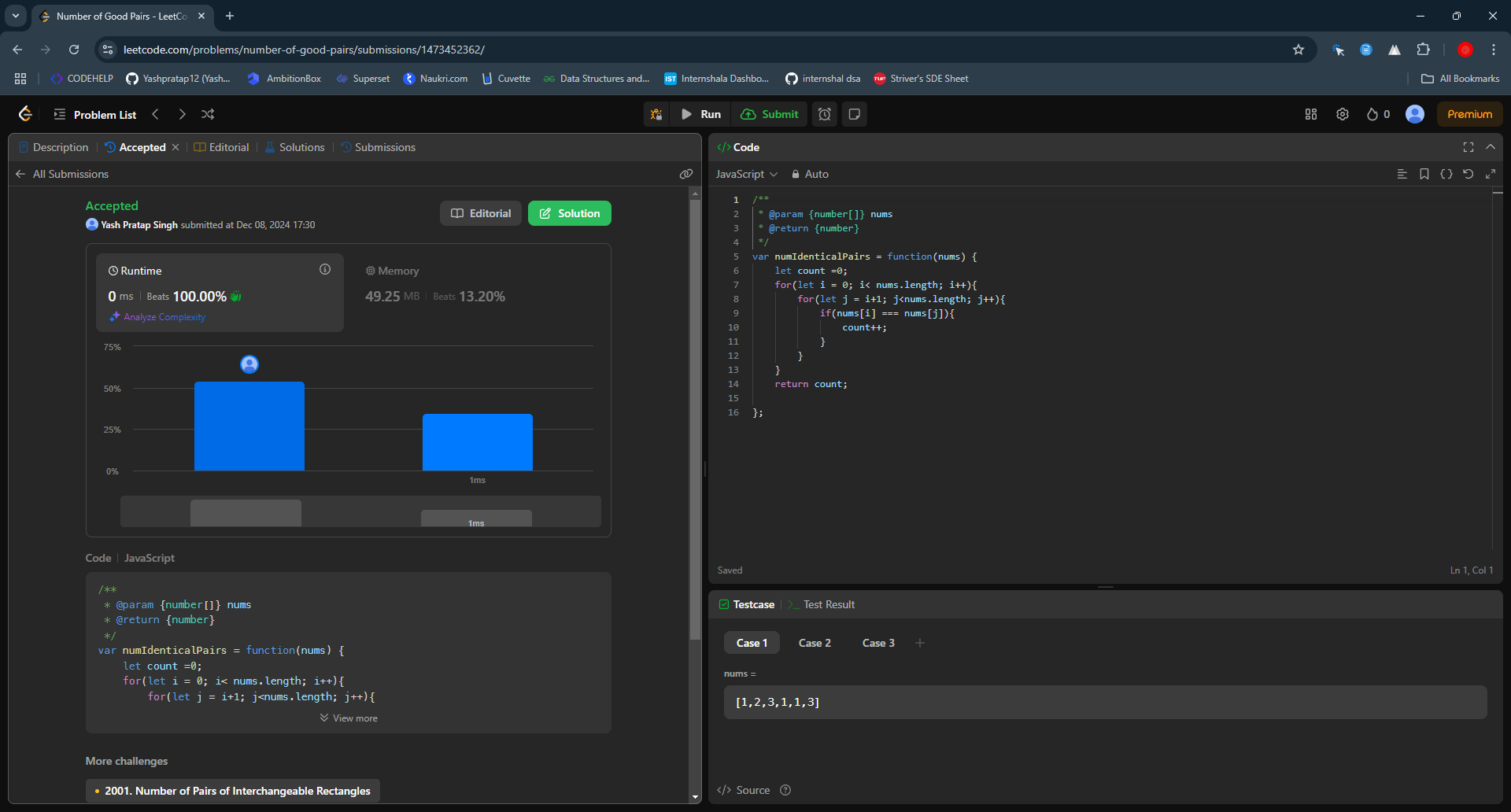
Leetcode 1512. https://leetcode.com/problems/number-of-good-pairs/description/

solution:- <https://leetcode.com/problems/number-of-good-pairs/submissions/1473452362/>

Time Complexity: O(n^2).

Space Complexity: O(1).

Added Screenshot.

.

Description:

There are two loop outer loop runs n times where n = nums.length means it will iterate over whole array. Inner loop will run on average (n-1)/2 times for each iteration of the outer loop. So the time complexity will become O(n^2).

Space complexity: No extra space required for the program execution so the complexity becomes O(1).

Question 5- Given the array nums consisting of 2n elements in the form [x1,x2,...,xn,y1,y2,...,yn]. Return the array in the form [x1,y1,x2,y2,...,xn,yn].

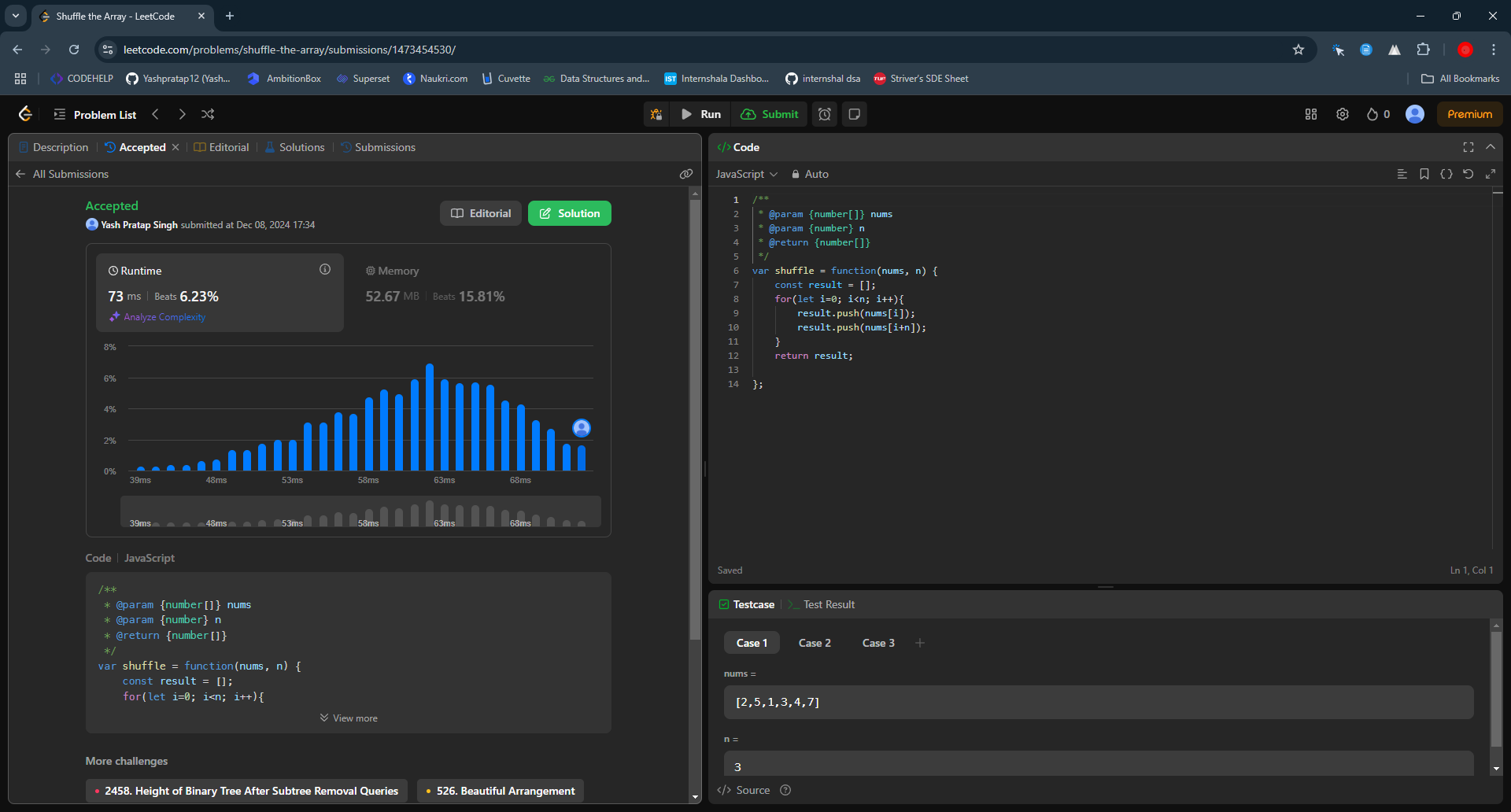
Leetcode 1470. https://leetcode.com/problems/shuffle-the-array/description/

solution:- <https://leetcode.com/problems/shuffle-the-array/submissions/1473454530/>

Time complexity: O(n).

Space complexity: O(n).

Added Screenshot.

.

Description:-

Time complexity: O(n)

outer loop iterates over the entire n- sized array so it will run n times. Each push operation take O(1)

and there is 2n operations for each x and y. So overall time complexity O(n).

Space Complexity: O(n) because array stores 2n elements which simplifies to O(n).