

DSA- Assignment-1

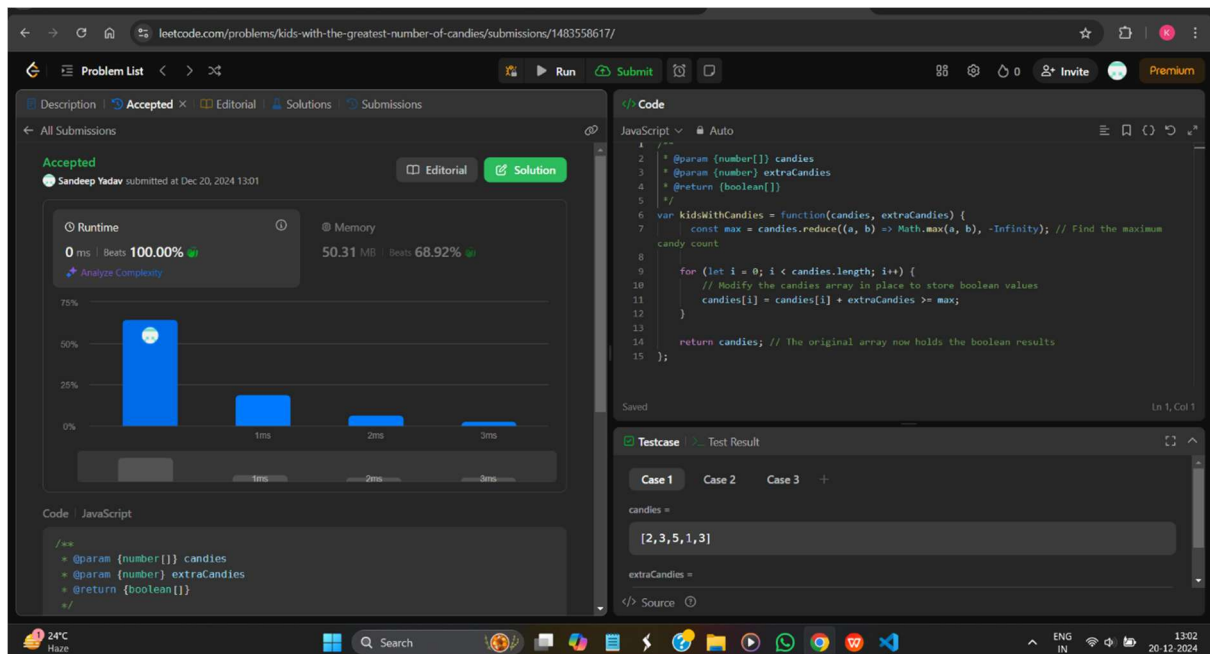
Questions List :-

1. Question was- <https://leetcode.com/problems/kids-with-the-greatest-number-of-candies/description/>

Solution Link –<https://leetcode.com/problems/kids-with-the-greatest-number-of-candies/submissions/1483558617/>

Time Complexity: $O(n)$

Space Complexity: $O(1)$

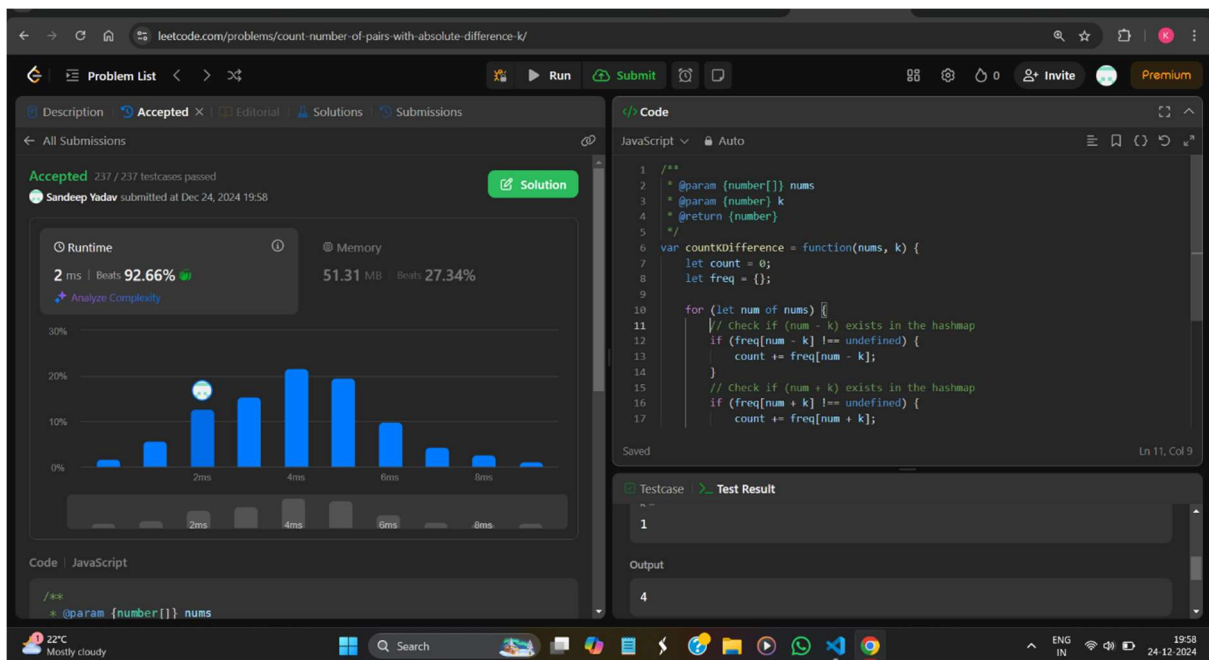


2. Question was- <https://leetcode.com/problems/count-number-of-pairs-with-absolute-difference-k/description/>

Solution Link– <https://leetcode.com/problems/count-number-of-pairs-with-absolute-difference-k/submissions/1487226106/>

Time Complexity: $O(n)$

Space Complexity: $O(n)$

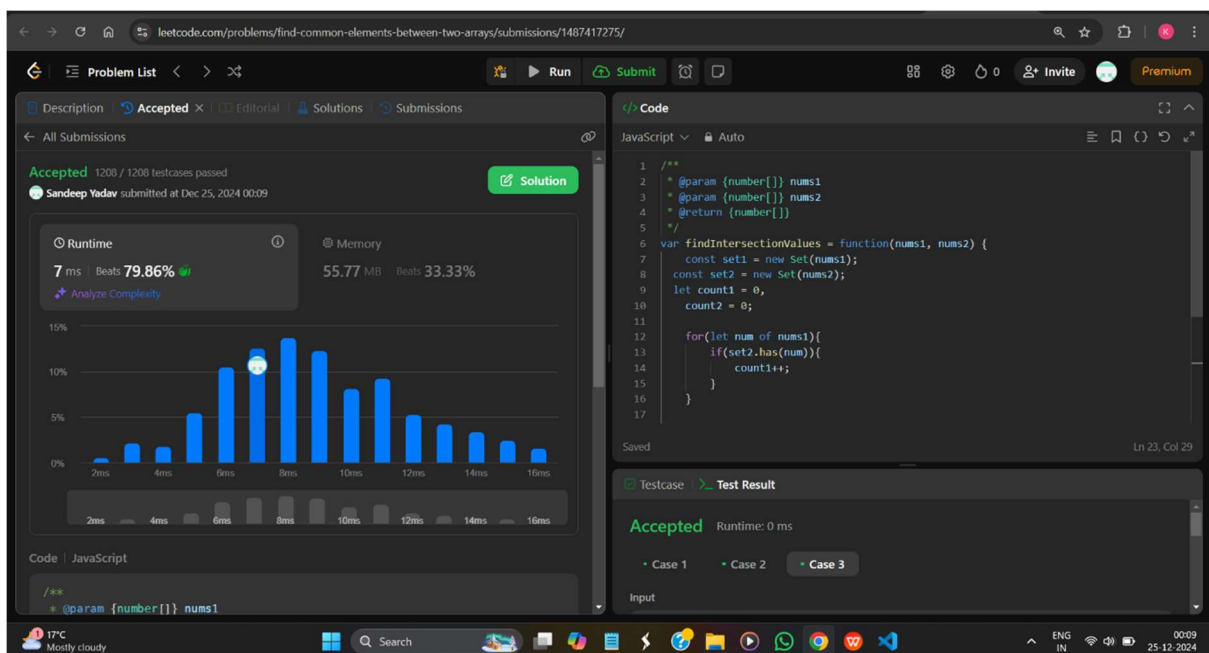


3. Question was- <https://leetcode.com/problems/find-common-elements-between-two-arrays/description/>

Solution Link– <https://leetcode.com/problems/find-common-elements-between-two-arrays/submissions/1487417275/>

Time Complexity: $O(n+m)$

Space Complexity: $O(n+m)$



4. Question was- <https://leetcode.com/problems/number-of-good-pairs/>

Solution Link- <https://leetcode.com/problems/number-of-good-pairs/submissions/1487459465/>

Time Complexity: $O(n)$

Space Complexity: $O(u)$ // u is number of unique elements

The screenshot displays the LeetCode submission page for the "Number of Good Pairs" problem. The submission is marked as "Accepted" with 49/49 test cases passed. The runtime is 0 ms, beating 100.00% of other submissions. The memory usage is 48.92 MB, beating 50.23%. The code is written in JavaScript and uses a frequency map to count the number of identical pairs in the array. The test result shows the expected output is 4.

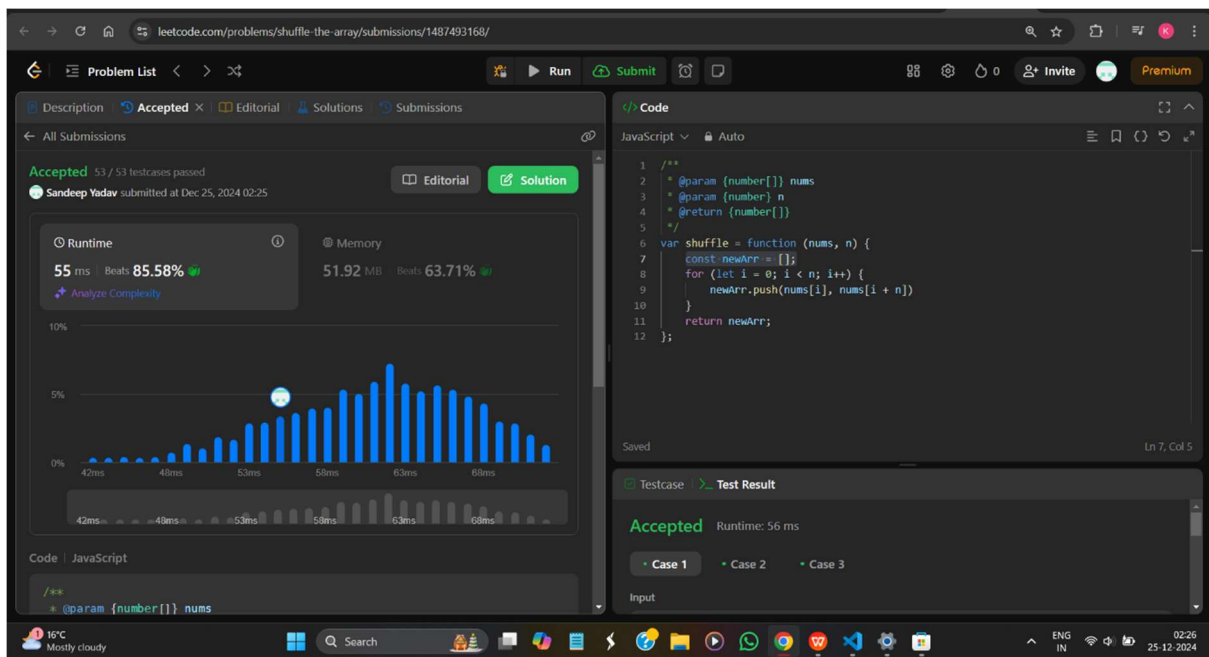
```
1 /**
2  * @param {number[]} nums
3  * @return {number}
4  */
5 var numIdenticalPairs = function(nums) {
6     const freqMap = new Map();
7     let count = 0;
8     const len = nums.length;
9     for(let i=0; i<len; i++){
10         if(freqMap.has(nums[i])){
11             count += freqMap.get(nums[i]);
12             freqMap.set(nums[i], freqMap.get(nums[i])+1)
13         } else{
14             freqMap.set(nums[i], 1)
15         }
16     }
17     // for(let i=0; i<len; i++){
18     }
```

5. Question was- <https://leetcode.com/problems/shuffle-the-array/>

Solution Link- <https://leetcode.com/problems/shuffle-the-array/submissions/1487493168/>

Time Complexity: $O(n)$

Space Complexity: $O(n)$



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.