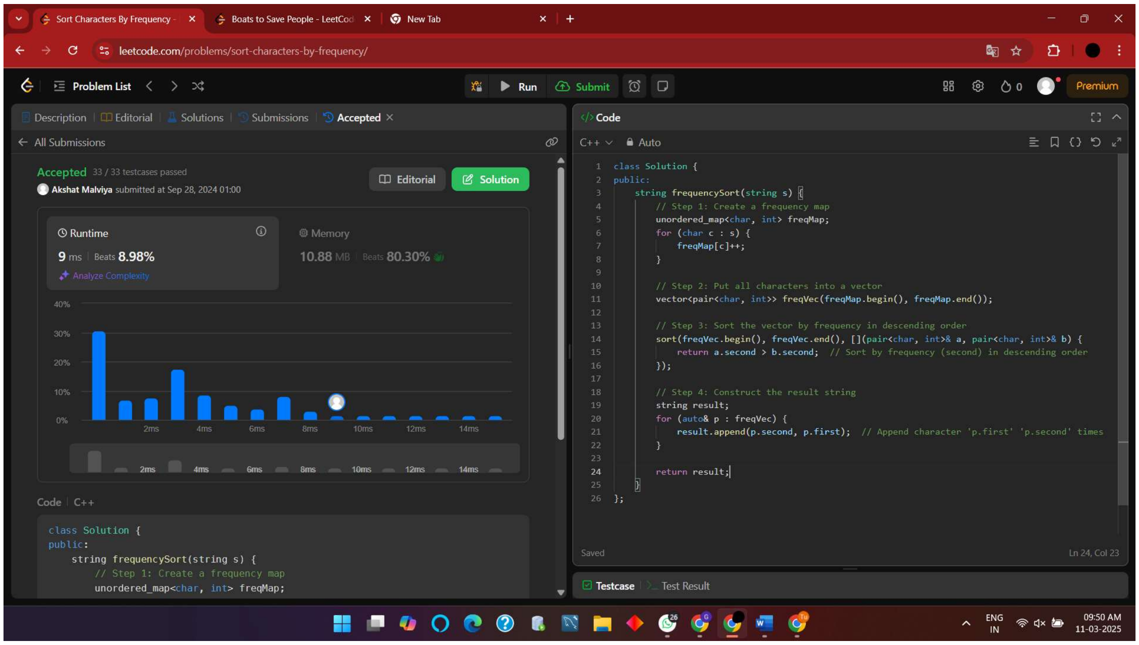
ASSIGNMENT 5 AP

Harsh K Sharma

22BCS16764

FL_IOT_603/A

451.Sort Characters By Frequency



Accepted 33 / 33 testcases passed

Akshat Malviya submitted at Sep 28, 2024 01:00

Runtime

9 ms | Beats 8.98%

Memory

10.88 MB | Beats 80.30%

Code | C++

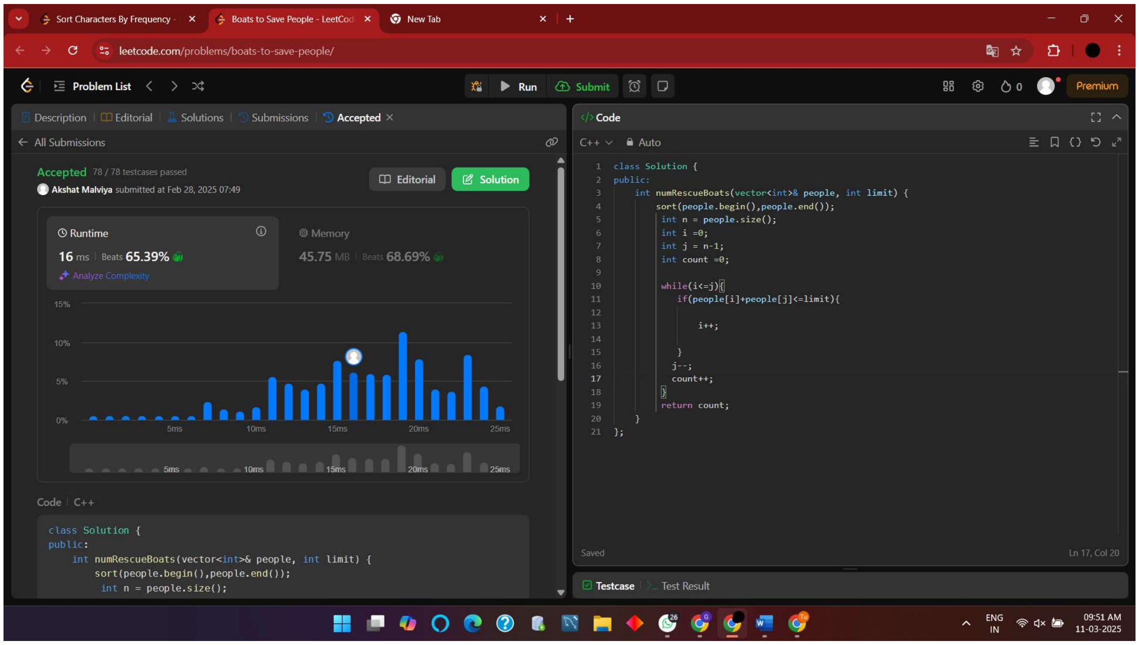
```cpp
class Solution {
public:
    string frequencySort(string s) {
        // Step 1: Create a frequency map
        unordered_map<char, int> freqMap;
```

```cpp
class Solution {
public:
    string frequencySort(string s) {
        // Step 1: Create a frequency map
        unordered_map<char, int> freqMap;
        for (char c : s) {
            freqMap[c]++;
        }

        // Step 2: Put all characters into a vector
        vector<pair<char, int>> freqVec(freqMap.begin(), freqMap.end());

        // Step 3: Sort the vector by frequency in descending order
        sort(freqVec.begin(), freqVec.end(), [](pair<char, int>& a, pair<char, int>& b) {
            return a.second > b.second;  // Sort by frequency (second) in descending order
        });

        // Step 4: Construct the result string
        string result;
        for (auto& p : freqVec) {
            result.append(p.second, p.first);  // Append character 'p.first' 'p.second' times
        }

        return result;
    }
};
```

Saved

Ln 24, Col 23

881.Boats to Save People



Accepted 78 / 78 testcases passed

Akshat Malviya submitted at Feb 28, 2025 07:49

Runtime 16 ms | Beats 65.39%

Memory 45.75 MB | Beats 68.69%

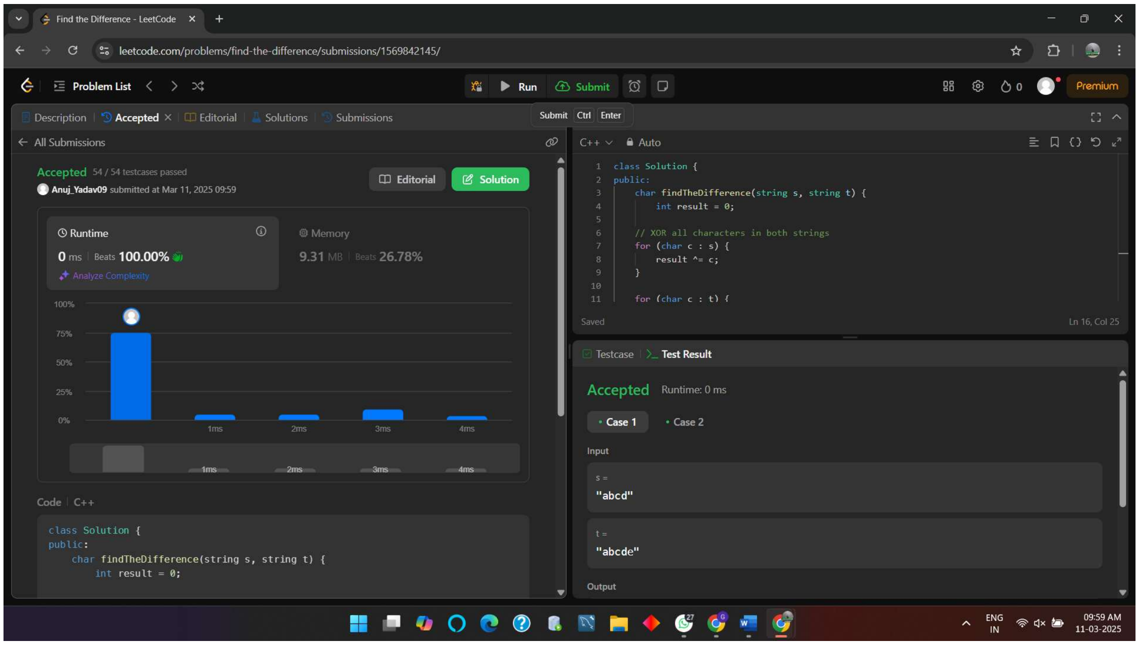
Code | C++

C++ Auto

```cpp
class Solution {
public:
    int numRescueBoats(vector<int>& people, int limit) {
        sort(people.begin(),people.end());
         int n = people.size();
         int i =0;
         int j = n-1;
         int count =0;

         while(i<=j){
            if(people[i]+people[j]<=limit){

                i++;

            }
           j--;
           count++;
         }
         return count;
    }
};
```

Saved   Ln 17, Col 20

389.[Find the diffrence](#)



Accepted 54 / 54 testcases passed

Anuj_Yadav09 submitted at Mar 11, 2025 09:59

Runtime 0 ms | Beats 100.00%

Memory 9.31 MB | Beats 26.78%

Code | C++

C++ Auto

```cpp
class Solution {
public:
    char findTheDifference(string s, string t) {
        int result = 0;

    // XOR all characters in both strings
    for (char c : s) {
        result ^= c;
    }

    for (char c : t) {
```

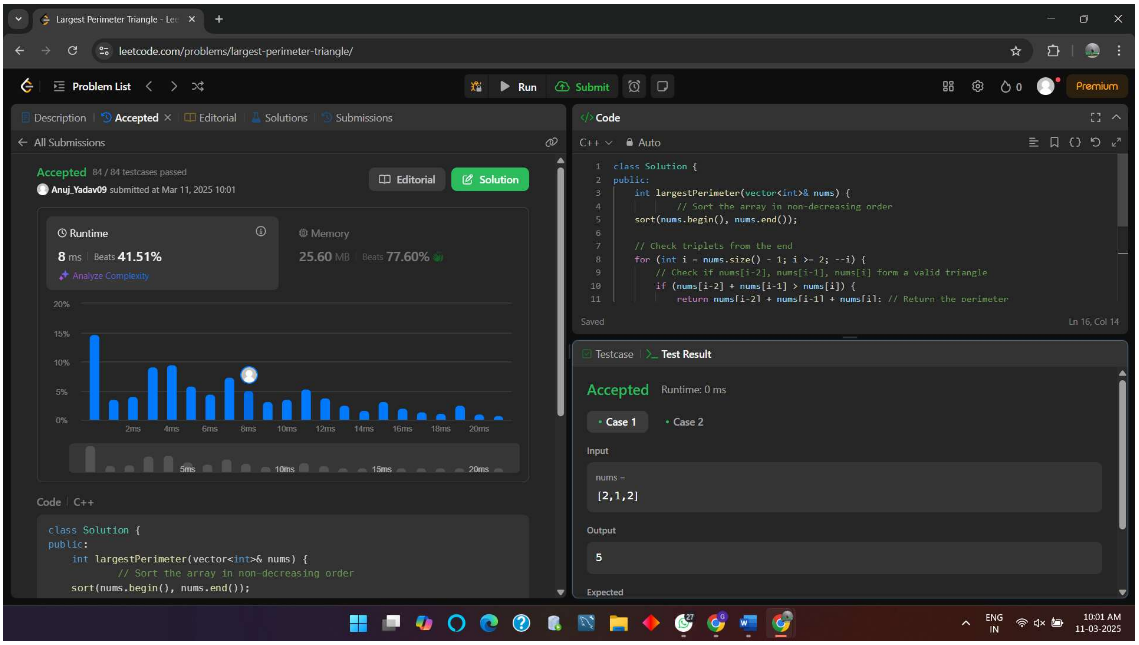
Saved   Ln 16, Col 25

Accepted Runtime: 0 ms

s =
"abcd"

t =
"abcde"

976.[Largest Perimeter Triangle](#)



Accepted 84 / 84 testcases passed

Anuj_Yadav09 submitted at Mar 11, 2025 10:01

Runtime 8 ms Beats 41.51%

Memory 25.60 MB Beats 77.60%

Code | C++

```
class Solution {
public:
    int largestPerimeter(vector<int>& nums) {
            // Sort the array in non-decreasing order
    sort(nums.begin(), nums.end());

    // Check triplets from the end
    for (int i = nums.size() - 1; i >= 2; --i) {
        // Check if nums[i-2], nums[i-1], nums[i] form a valid triangle
        if (nums[i-2] + nums[i-1] > nums[i]) {
            return nums[i-2] + nums[i-1] + nums[i]; // Return the perimeter
```

Saved Ln 16, Col 14

Accepted Runtime: 0 ms

Input

nums =

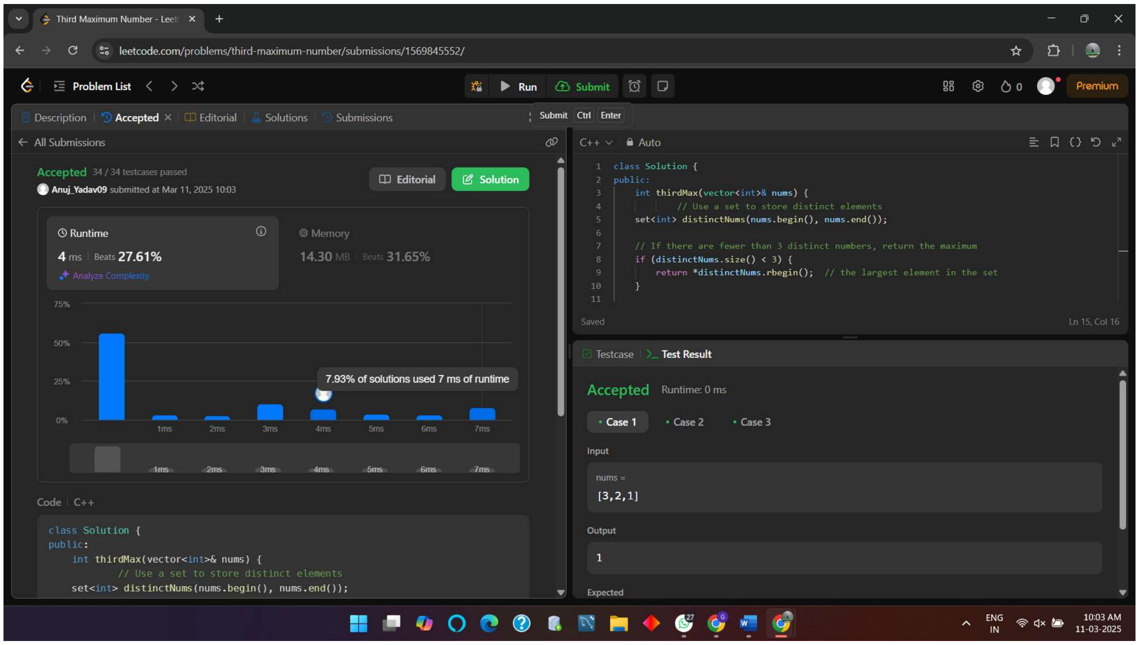
[2,1,2]

Output

5

Expected

414.[Third Maximum Number]



Accepted 34 / 34 testcases passed

Anuj_Yadav09 submitted at Mar 11, 2025 10:03

Runtime 4 ms Beats 27.61%

Memory 14.30 MB Beats 31.65%

Code | C++

```
class Solution {
public:
    int thirdMax(vector<int>& nums) {
            // Use a set to store distinct elements
    set<int> distinctNums(nums.begin(), nums.end());

    // If there are fewer than 3 distinct numbers, return the maximum
    if (distinctNums.size() < 3) {
        return *distinctNums.rbegin();  // the largest element in the set
    }

```

Saved Ln 15, Col 16

Accepted Runtime: 0 ms

Input

nums =

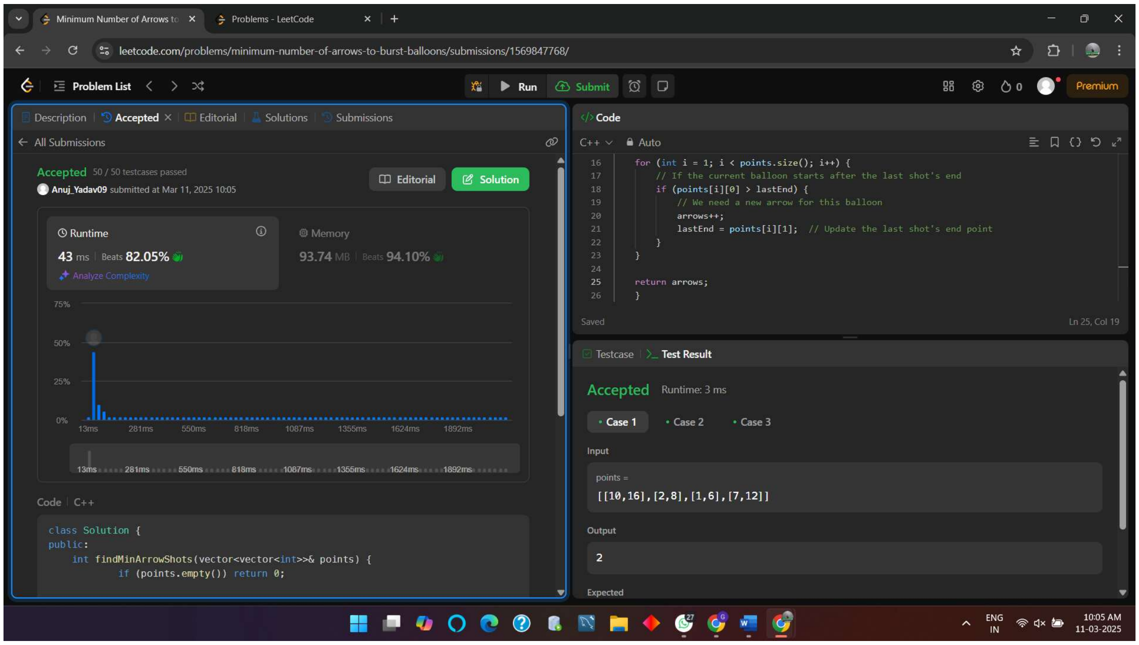
[3,2,1]

Output

1

Expected

452.[Minimum Number of Arrows to Burst Balloons]



Accepted 50 / 50 testcases passed

Anuj_Yadav09 submitted at Mar 11, 2025 10:05

Runtime 43 ms | Beats 82.05%

Memory 93.74 MB | Beats 94.10%

Code | C++

```cpp
class Solution {
public:
    int findMinArrowShots(vector<vector<int>>& points) {
        if (points.empty()) return 0;
```

```cpp
    for (int i = 1; i < points.size(); i++) {
        // If the current balloon starts after the last shot's end
        if (points[i][0] > lastEnd) {
            // We need a new arrow for this balloon
            arrows++;
            lastEnd = points[i][1];  // Update the last shot's end point
        }
    }

    return arrows;
    }
```

Accepted Runtime: 3 ms

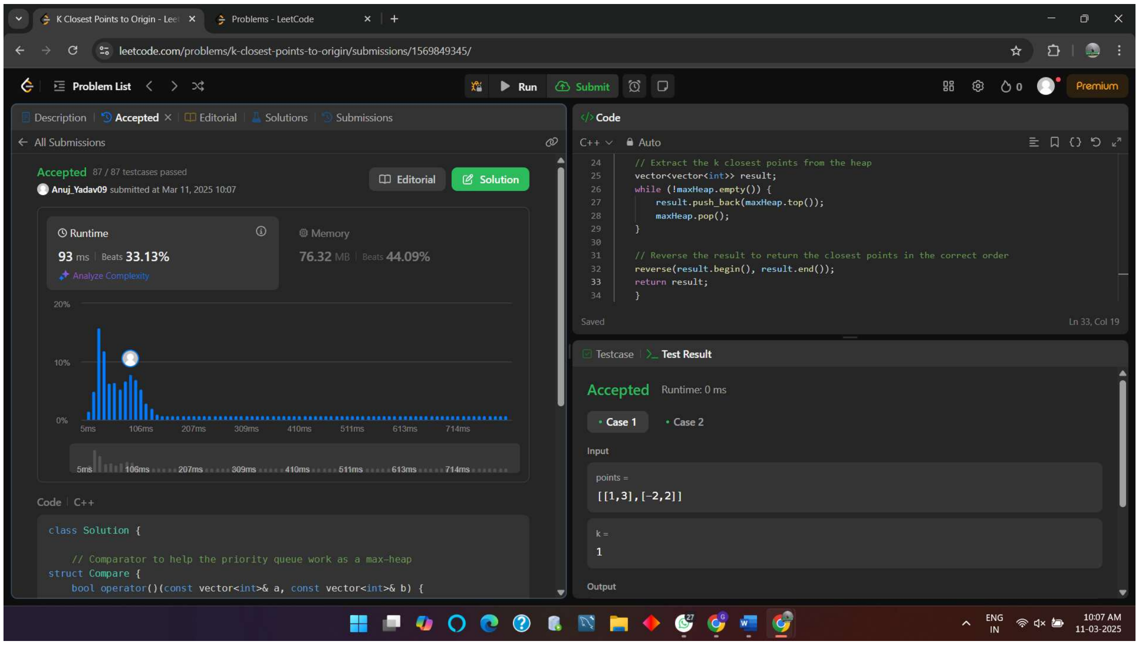
points =

```
[[10,16],[2,8],[1,6],[7,12]]
```

Output

```
2
```

973.K Closest Points to Origin



Accepted 87 / 87 testcases passed

Anuj_Yadav09 submitted at Mar 11, 2025 10:07

Runtime 93 ms | Beats 33.13%

Memory 76.32 MB | Beats 44.09%

Code | C++

```cpp
class Solution {

    // Comparator to help the priority queue work as a max-heap
struct Compare {
    bool operator()(const vector<int>& a, const vector<int>& b) {
```

```cpp
    // Extract the k closest points from the heap
    vector<vector<int>> result;
    while (!maxHeap.empty()) {
        result.push_back(maxHeap.top());
        maxHeap.pop();
    }

    // Reverse the result to return the closest points in the correct order
    reverse(result.begin(), result.end());
    return result;
    }
```

Accepted Runtime: 0 ms

points =

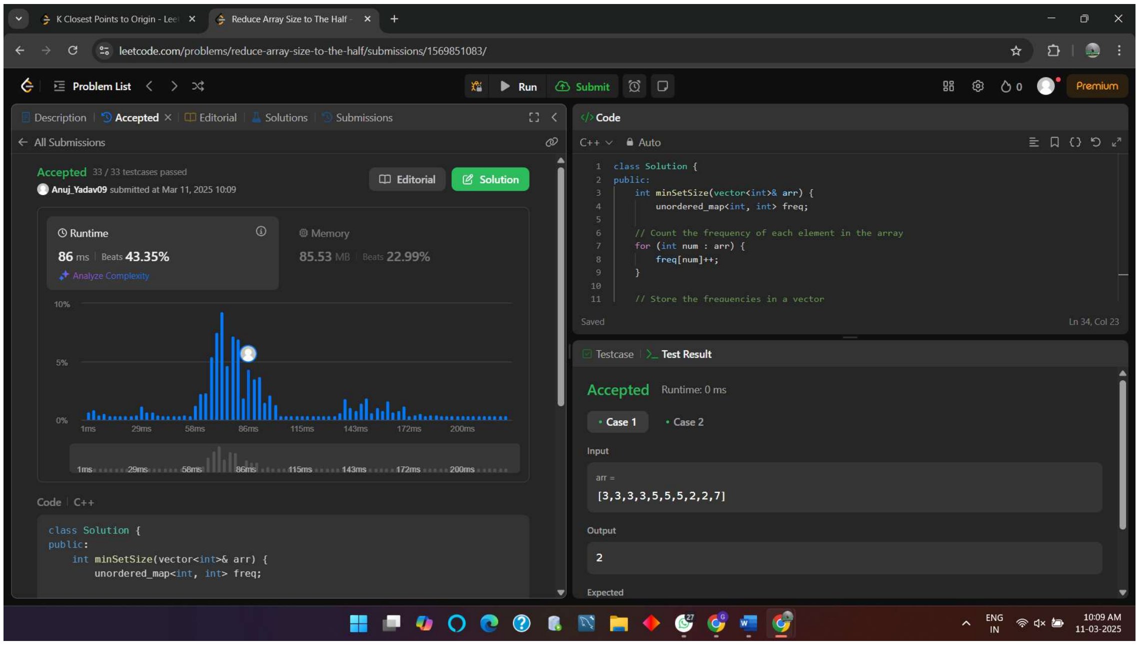
```
[[1,3],[-2,2]]
```

k =

```
1
```

Output

1338.Reduce Array Size to The Half



Accepted 33 / 33 testcases passed

Anuj_Yadav09 submitted at Mar 11, 2025 10:09

Runtime

86 ms | Beats 43.35%

Analyze Complexity

Memory

85.53 MB | Beats 22.99%

Code | C++

```
class Solution {
public:
    int minSetSize(vector<int>& arr) {
        unordered_map<int, int> freq;
```

C++ Auto

```
class Solution {
public:
    int minSetSize(vector<int>& arr) {
        unordered_map<int, int> freq;

    // Count the frequency of each element in the array
    for (int num : arr) {
        freq[num]++;
    }

    // Store the frequencies in a vector
```

Saved

Ln 34, Col 23

Accepted Runtime: 0 ms

Input

arr =

[3,3,3,3,5,5,5,2,2,7]

Output

2

Expected

10:09 AM
11-03-2025