

Advanced Programming LAB II

ASSIGNMENT - 5

Submitted by,

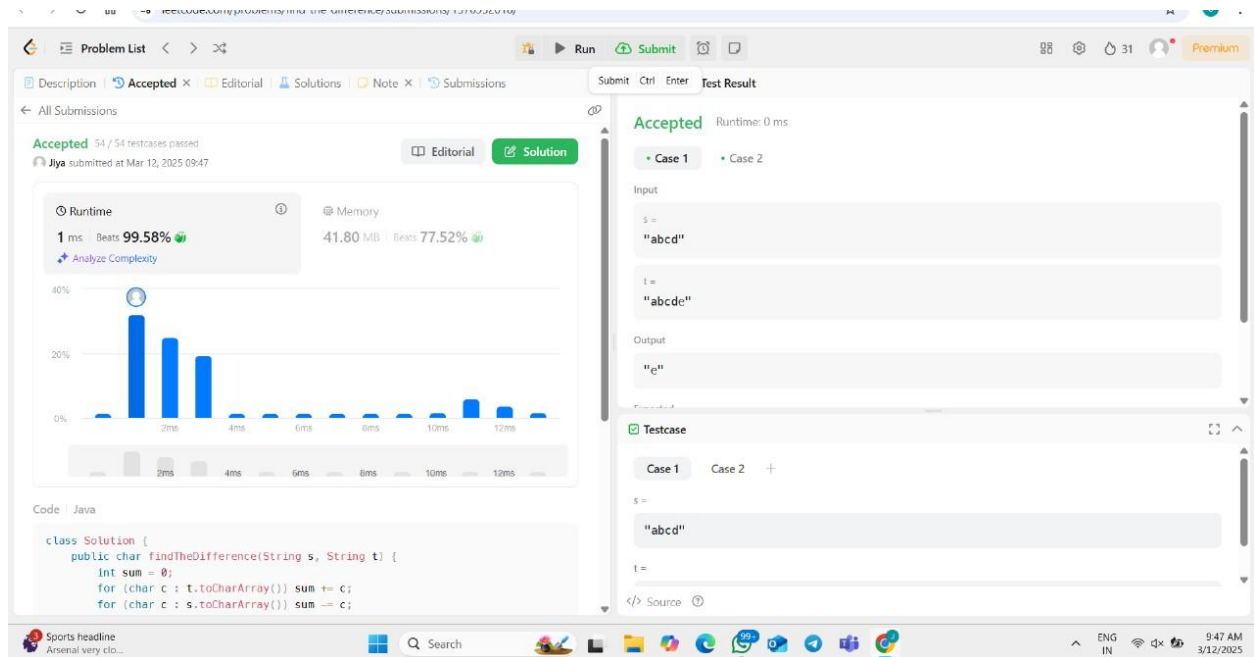
Jiya | 22BCS14856

22BCS_FL_IOT-601 (A)

389. Find the difference

<https://leetcode.com/problems/find-the-difference/description/>

```
class Solution {  
    public char findTheDifference(String s, String t) {  
        int sum = 0;  
        for (char c : t.toCharArray()) sum += c;  
        for (char c : s.toCharArray()) sum -= c;  
        return (char) sum;  
    }  
}
```

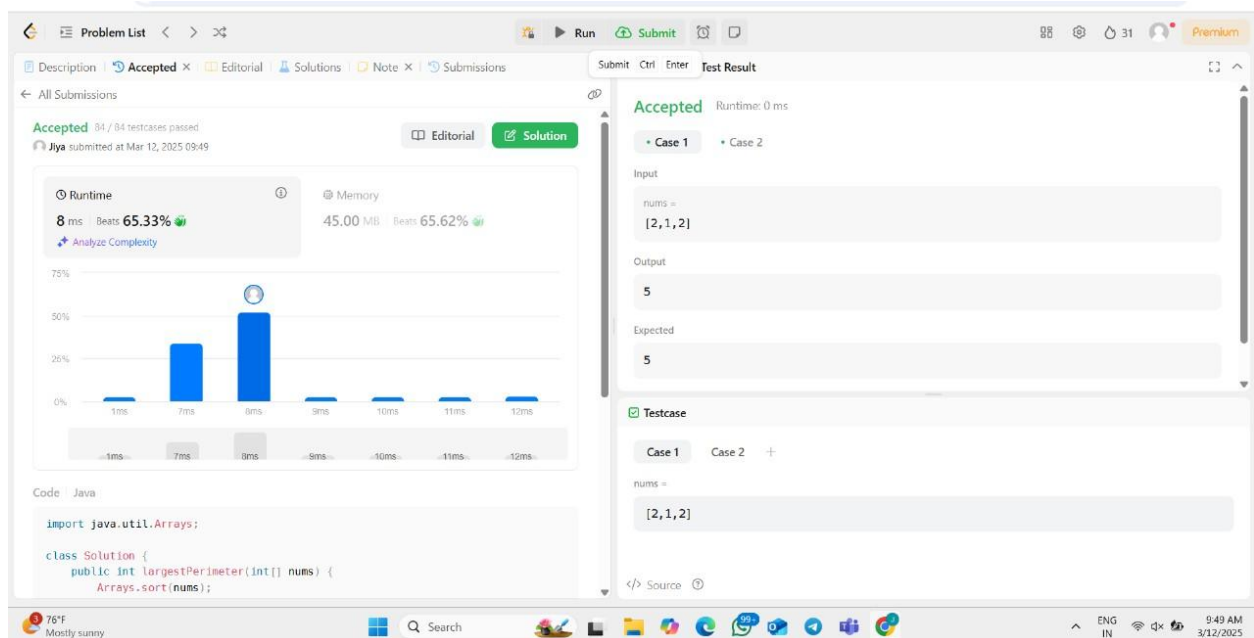


976. Largest Perimeter Triangle

<https://leetcode.com/problems/largest-perimeter-triangle/description/>

```
import java.util.Arrays;

class Solution {
    public int largestPerimeter(int[] nums) {
        Arrays.sort(nums);
        for (int i = nums.length - 1; i >= 2; i--) {
            if (nums[i - 1] + nums[i - 2] > nums[i]) {
                return nums[i] + nums[i - 1] + nums[i - 2];
            }
        }
        return 0;
    }
}
```

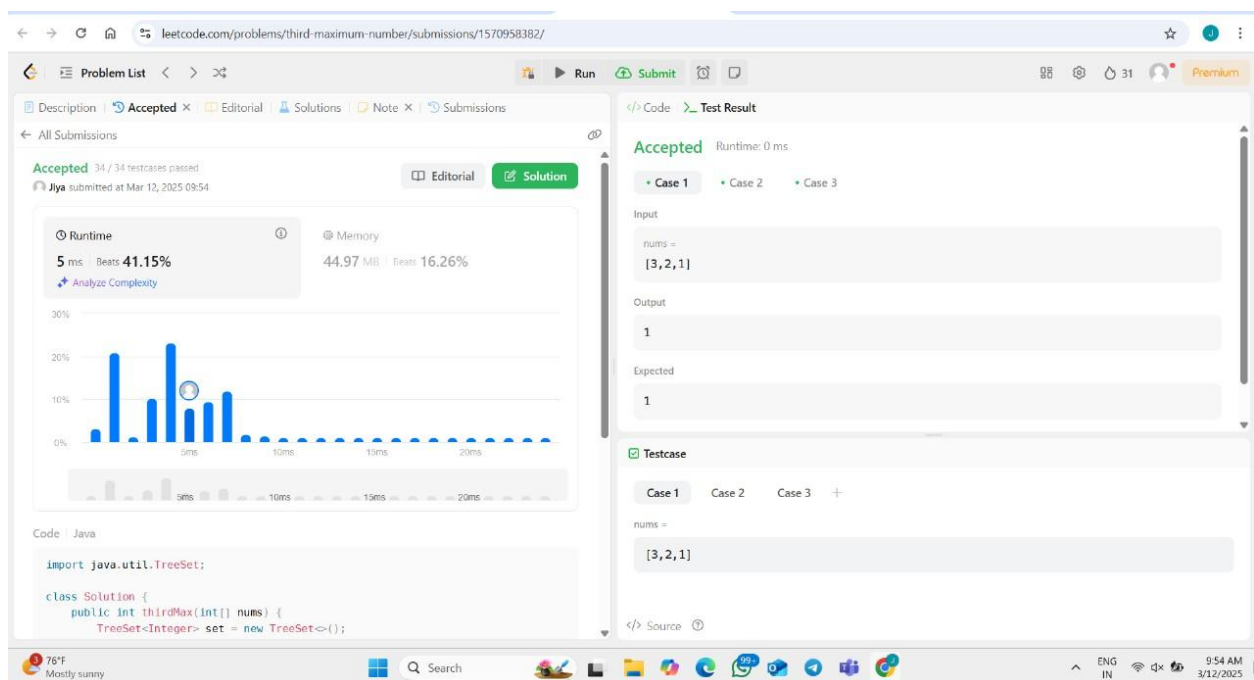


414. Third Maximum Number

<https://leetcode.com/problems/third-maximum-number/description/>

```
import java.util.TreeSet;

class Solution {
    public int thirdMax(int[] nums) {
        TreeSet<Integer> set = new TreeSet<>();
        for (int num : nums) {
            set.add(num);
            if (set.size() > 3) set.pollFirst();
        }
        return set.size() == 3 ? set.first() : set.last();
    }
}
```



451. Sort Characters By Frequency

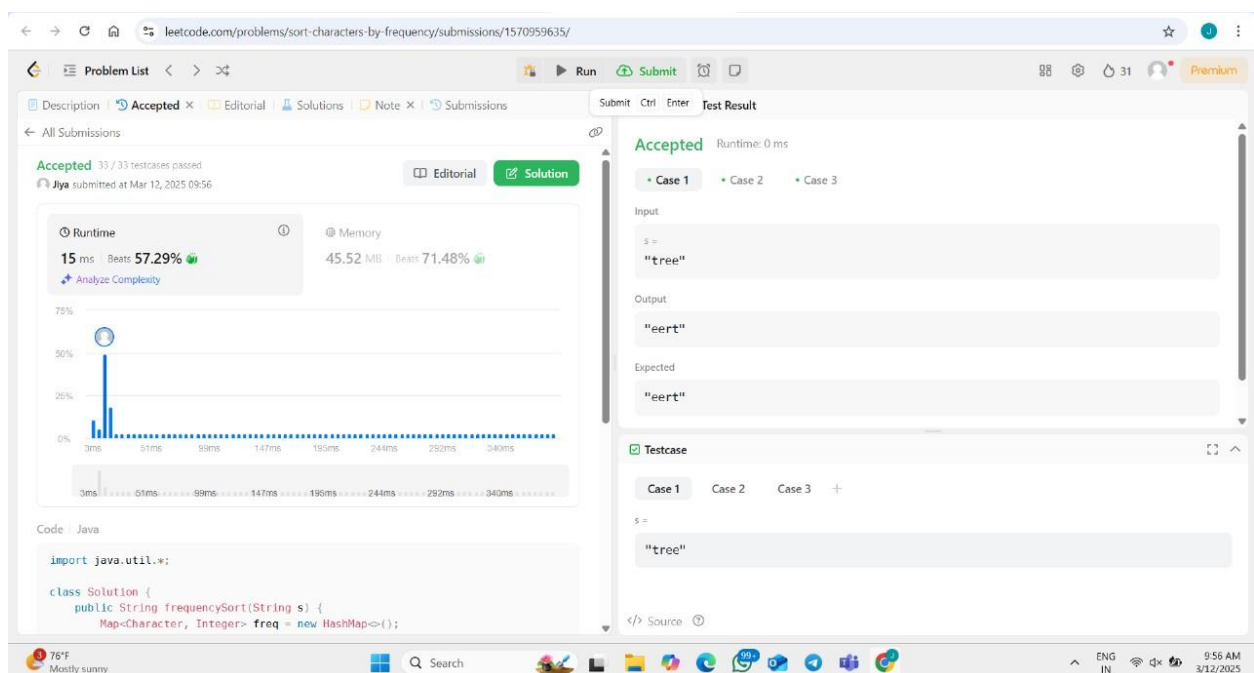
<https://leetcode.com/problems/sort-characters-by-frequency/description/>

```
import java.util.*;

class Solution {
    public String frequencySort(String s) {
        Map<Character, Integer> freq = new HashMap<>();
        for (char c : s.toCharArray()) freq.put(c, freq.getOrDefault(c, 0) + 1);

        PriorityQueue<Character> pq = new PriorityQueue<>((a, b) -> freq.get(b) -
freq.get(a));
        pq.addAll(freq.keySet());

        StringBuilder res = new StringBuilder();
        while (!pq.isEmpty()) {
            char c = pq.poll();
            res.append(String.valueOf(c).repeat(freq.get(c)));
        }
        return res.toString();
    }
}
```

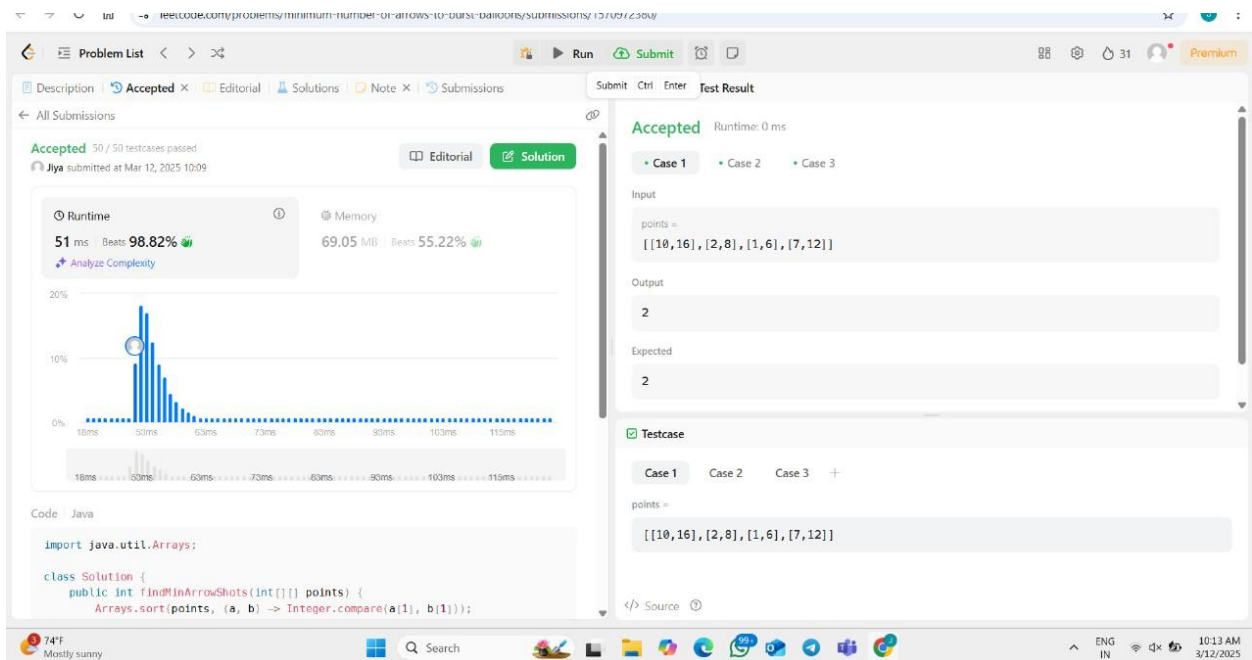


452.Minimum Number of Arrows to Burst Balloons

<https://leetcode.com/problems/minimum-number-of-arrows-to-burst-balloons/>

```
import java.util.Arrays;

class Solution {
    public int findMinArrowShots(int[][] points) {
        Arrays.sort(points, (a, b) -> Integer.compare(a[1], b[1]));
        int arrows = 1, end = points[0][1];
        for (int i = 1; i < points.length; i++) {
            if (points[i][0] > end) {
                arrows++;
                end = points[i][1];
            }
        }
        return arrows;
    }
}
```

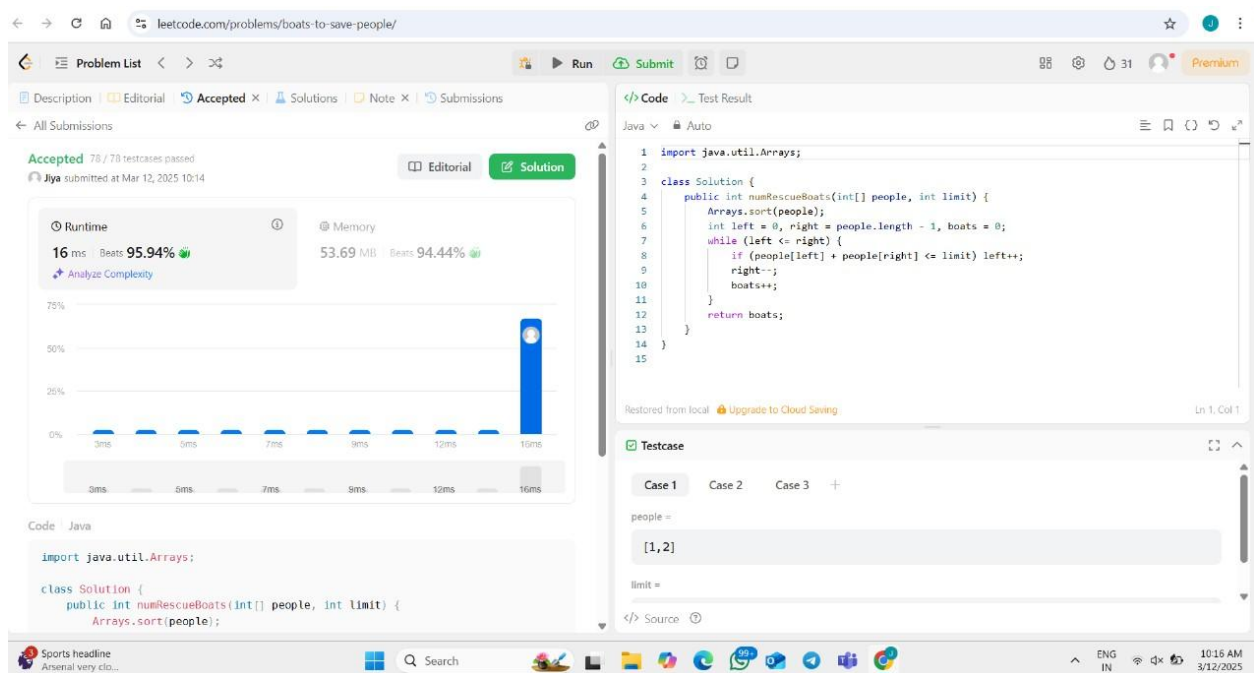


881. Boats to Save People

<https://leetcode.com/problems/boats-to-save-people/description/>

```
import java.util.Arrays;

class Solution {
    public int numRescueBoats(int[] people, int limit) {
        Arrays.sort(people);
        int left = 0, right = people.length - 1, boats = 0;
        while (left <= right) {
            if (people[left] + people[right] <= limit) left++;
            right--;
            boats++;
        }
        return boats;
    }
}
```



973. K Closest Points to Origin

<https://leetcode.com/problems/k-closest-points-to-origin/description/>

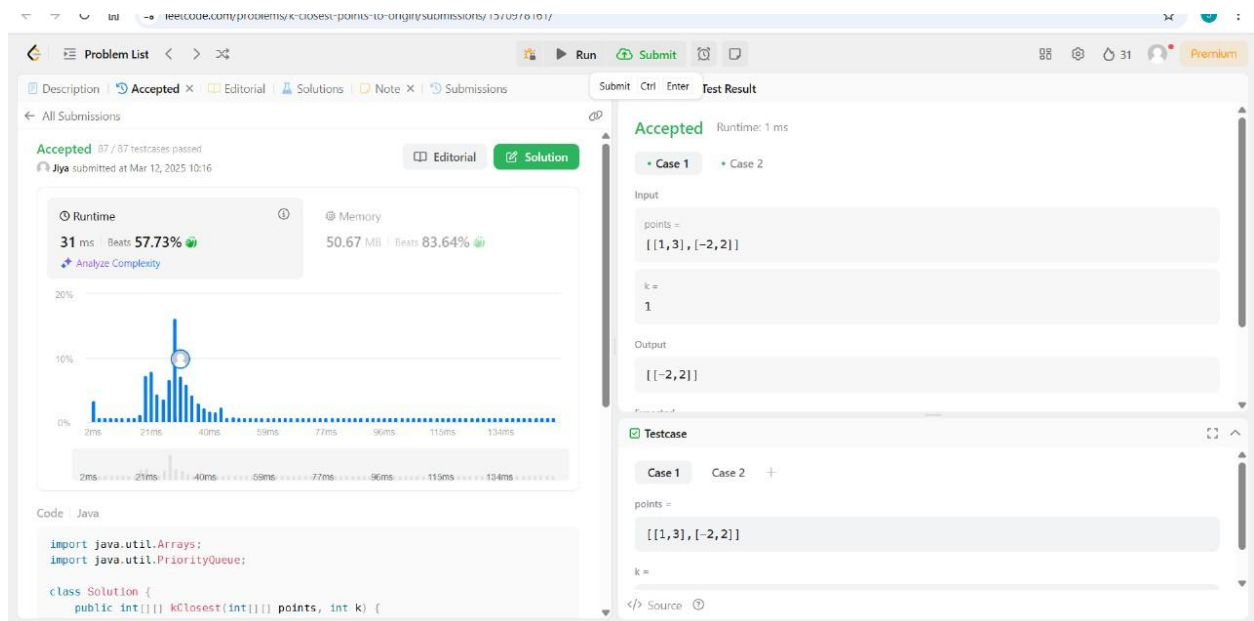
```
import java.util.Arrays;
import java.util.PriorityQueue;

class Solution {
    public int[][] kClosest(int[][] points, int k) {
        PriorityQueue<int[]> pq = new PriorityQueue<>((a, b) ->
            (b[0] * b[0] + b[1] * b[1]) - (a[0] * a[0] + a[1] * a[1]));

        for (int[] p : points) {
            pq.offer(p);
            if (pq.size() > k) pq.poll();
        }

        int[][] res = new int[k][2];
        while (k-- > 0) res[k] = pq.poll();

        return res;
    }
}
```



1338. Reduce Array Size to The Half

<https://leetcode.com/problems/reduce-array-size-to-the-half/description/>

```
import java.util.*;

class Solution {
    public int minSetSize(int[] arr) {
        Map<Integer, Integer> freqMap = new HashMap<>();
        for (int num : arr) {
            freqMap.put(num, freqMap.getOrDefault(num, 0) + 1);
        }

        List<Integer> freqList = new ArrayList<>(freqMap.values());
        Collections.sort(freqList, Collections.reverseOrder());

        int halfSize = arr.length / 2;
        int removed = 0, count = 0;

        for (int freq : freqList) {
            removed += freq;
            count++;
            if (removed >= halfSize) break;
        }

        return count;
    }
}
```

leetcode.com/problems/reduce-array-size-to-the-half/submissions/1570980515/

Problem List < > Run Submit Test Result

Description Accepted x Editorial Solutions Note x Submissions

All Submissions

Accepted 31 / 33 testcases passed

jiye submitted at Mar 12, 2025 10:18

Editorial Solution

Runtime 36 ms Beats 61.31%

Memory 60.92 MB Beats 55.72%

Analyze Complexity

10% 5% 0% 3ms 15ms 20ms 30ms 40ms

Code: Java

```
import java.util.*;

class Solution {
    public int minSetSize(int[] arr) {
        Map<Integer, Integer> freqMap = new HashMap<>();
```

Accepted Runtime: 0 ms

Case 1 Case 2

Input

arr = [3, 3, 3, 3, 5, 5, 2, 2, 7]

Output

2

Expected

2

Testcase

Case 1 Case 2 +

arr = [3, 3, 3, 3, 5, 5, 2, 2, 7]

</> Source

Sports headline Arsenal very clo...

Search

ENG IN 10:18 AM 3/12/2025