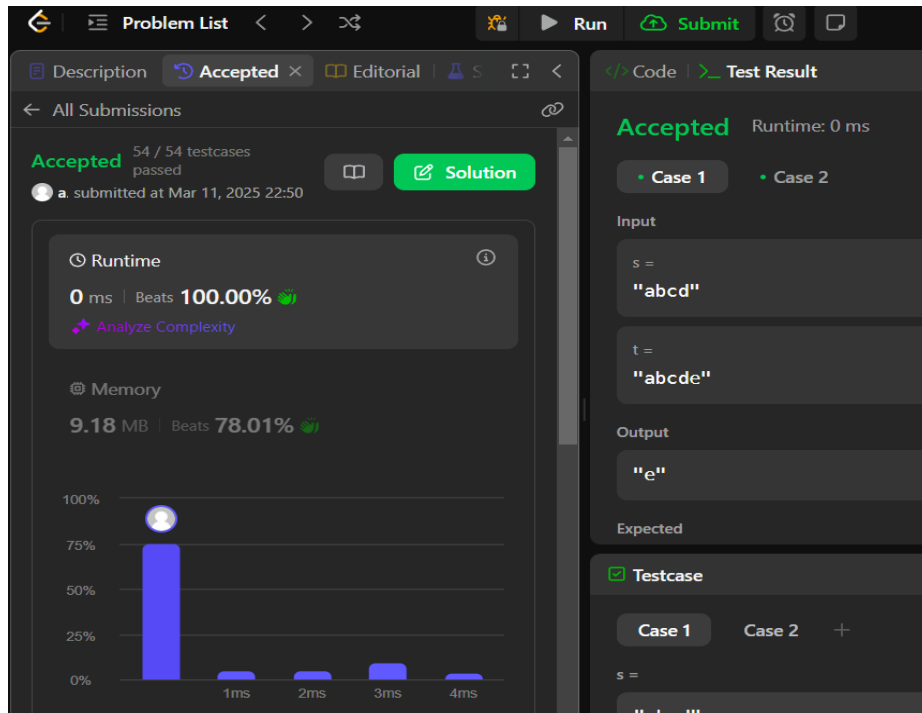
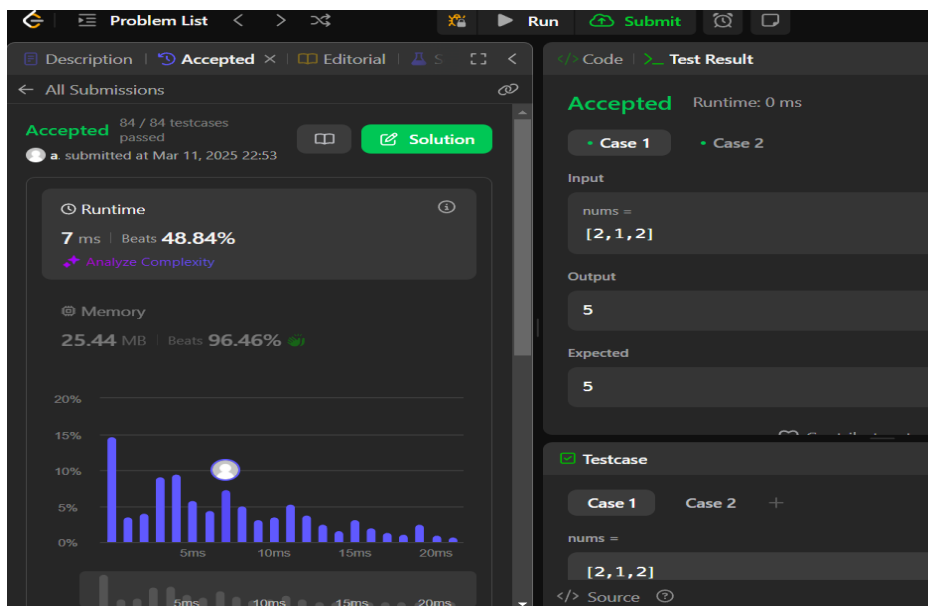


AP 5

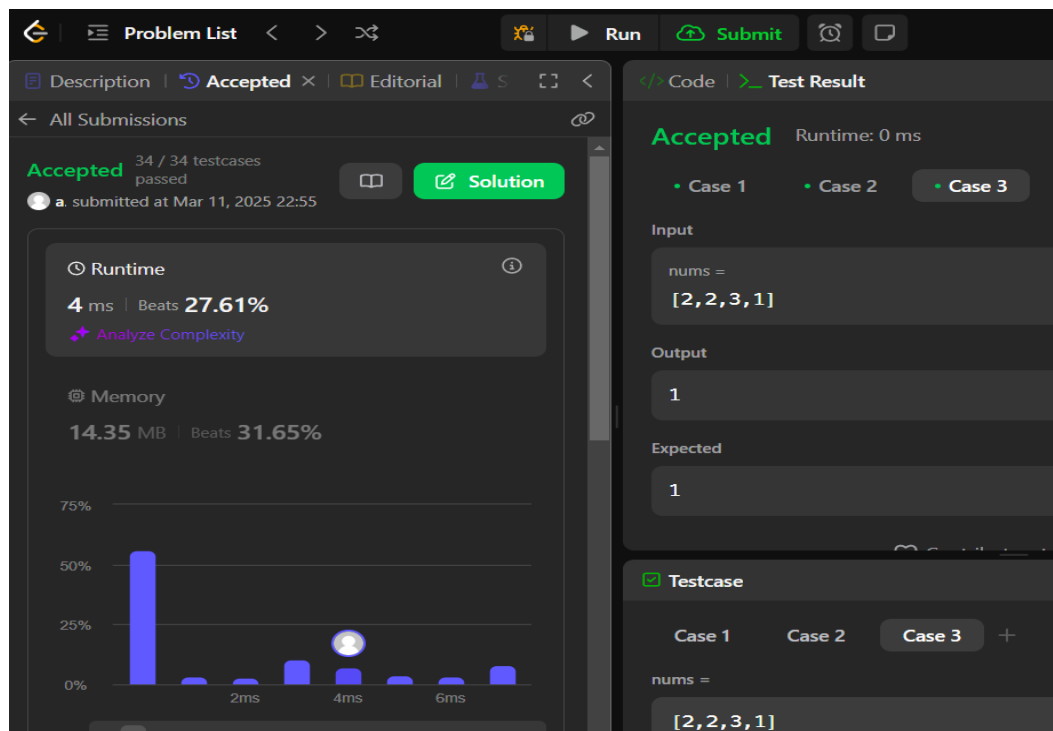
389.[Find the difference](#)



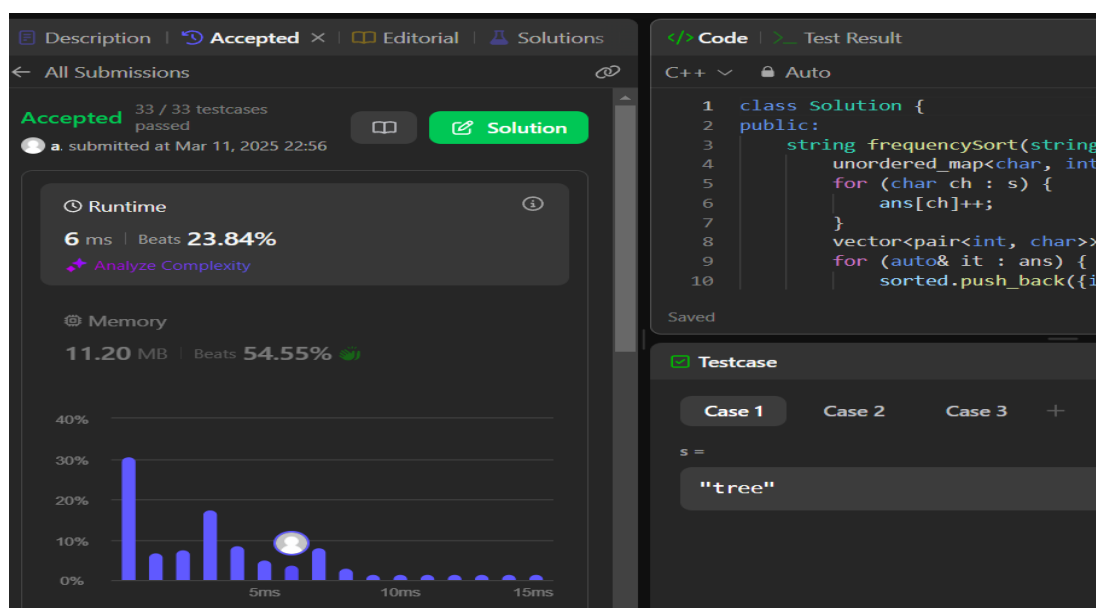
976.[Largest Perimeter Triangle](#)



414. [Third Maximum Number](#)



451. [Sort Characters By Frequency](#)



452. [Minimum Number of Arrows to Burst Balloons](#)

Description | **Accepted** | Editorial | Solutions

All Submissions

Accepted 50 / 50 testcases passed

a. submitted at Mar 11, 2025 22:58

Runtime
36 ms | Beats 95.78%
[Analyze Complexity](#)

Memory
93.91 MB | Beats 49.20%

75%
50%
25%
0%

13ms 684ms 1355ms 2027ms

Code | Test Result

C++ | Auto

```
20     prevEnd = balloon  
21     position  
22     }  
23  
24     return arrows;  
25 }  
26 }  
27
```

Saved

Testcase

Case 1 Case 2 Case 3 +

points =

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

881. [Boats to Save People](#)

Problem List | **Accepted** | Editorial | Solutions

All Submissions

Accepted 78 / 78 testcases passed

a. submitted at Mar 11, 2025 22:59

Runtime
11 ms | Beats 92.17%
[Analyze Complexity](#)

Memory
45.80 MB | Beats 68.69%

15%
10%
5%
0%

10ms 20ms

Code | Test Result

C++ | Auto

```
14     }  
15     right--; // Heavies  
16     boats++; // Count t  
17 }  
18  
19     return boats;  
20 }  
21 }  
22
```

Saved

Testcase

Case 1 Case 2 Case 3 +

people =

[1,2]

limit =

3

</> Source ?

973. K Closest Points to Origin

The screenshot shows a LeetCode problem solution for "K Closest Points to Origin". The interface is in dark mode. On the left, the "Runtime" section shows 86 ms and 38.65% beats, while the "Memory" section shows 77.99 MB and 39.71% beats. A histogram below these metrics shows the distribution of runtime and memory usage across other submissions. The top navigation bar includes "Problem List", "Run", "Submit", and "Premi". The main content area is divided into three sections: "Code", "Testcase", and "Test Result". The "Code" section contains the following C++ code:

```
1 #include <vector>
2 #include <queue>
3
4 class Solution {
5 public:
6     std::vector<std::vector<int>>> kClosest
7     (std::vector<std::vector<int>>>& points, int k) {
8         // Max heap to store the k closest points
9         std::priority_queue<std::pair<int, std::vector<int>>>>
10         maxHeap;
```

The "Testcase" section shows "Case 1" with the input `points = [[1,3], [-2,2]]` and `k = 1`. The "Test Result" section is currently empty.