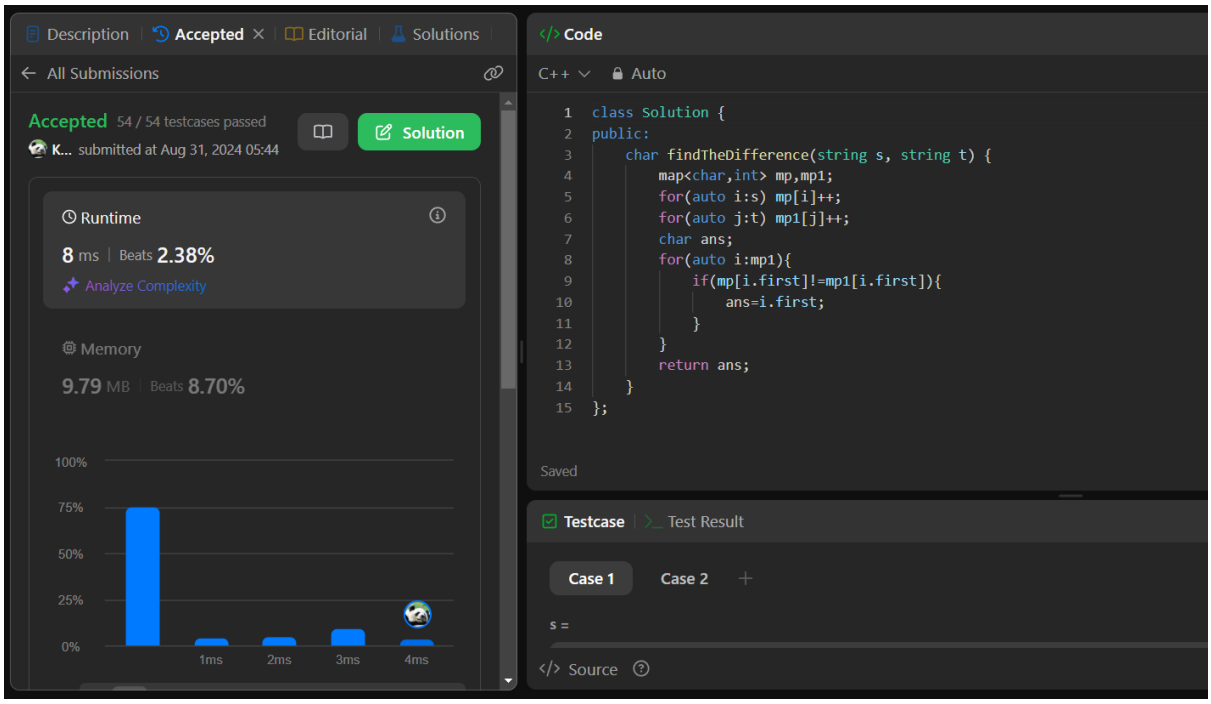
Name: Krishna Sharma

UID: 22BCS11885

Section: FL-IOT-641-A

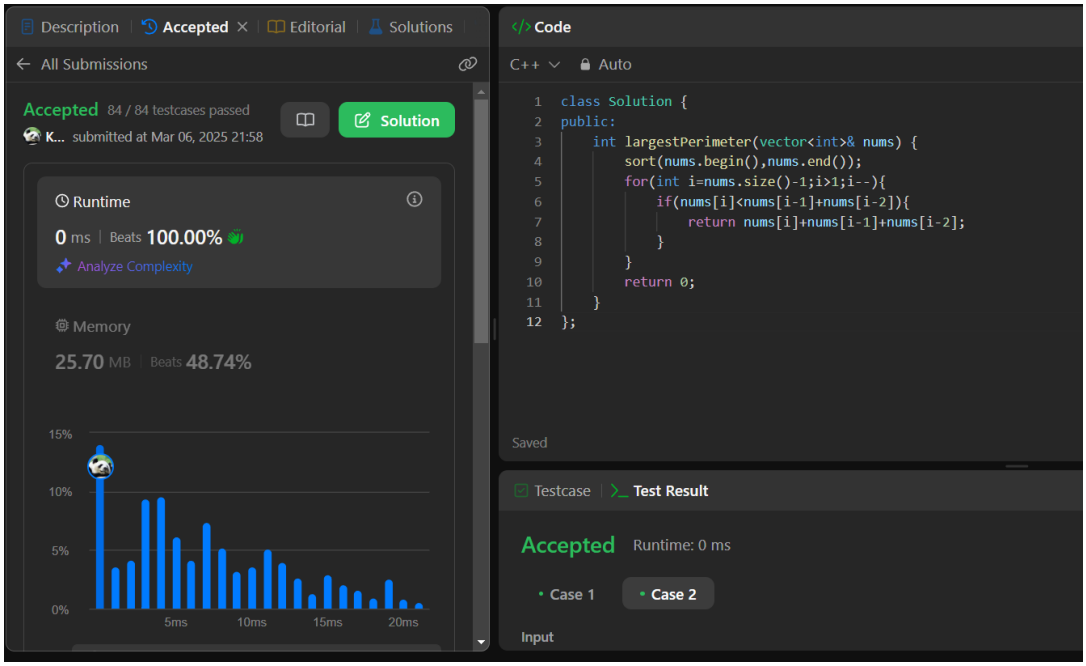
389. Find the Difference

```
class Solution {
public:
   char findTheDifference(string s, string t) {
      map<char,int> mp,mp1;
      for(auto i:s) mp[i]++;
      for(auto j:t) mp1[j]++;
      char ans;
      for(auto i:mp1){
         if(mp[i.first]!=mp1[i.first]){
            ans=i.first;
         }
      }
      return ans;
   }
};
```



976. Largest Perimeter Triangle

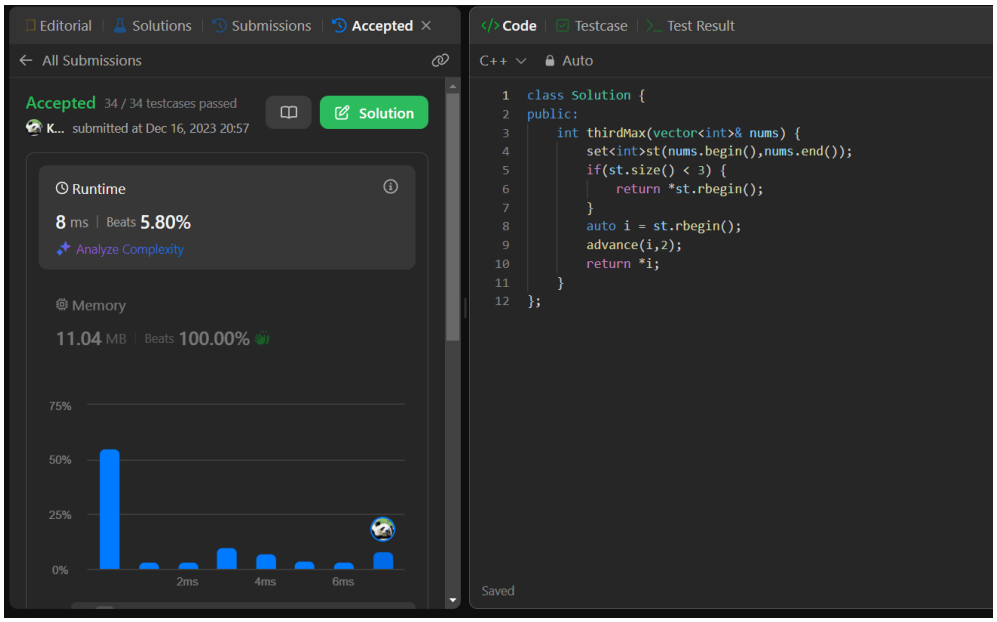
```
class Solution {
public:
    int largestPerimeter(vector<int>& nums) {
        sort(nums.begin(),nums.end());
        for(int i=nums.size()-1;i>1;i--){
            if(nums[i]<nums[i-1]+nums[i-2]){
                return nums[i]+nums[i-1]+nums[i-2];
            }
        }
        return 0;
    }
};
```



414. Third Maximum Number

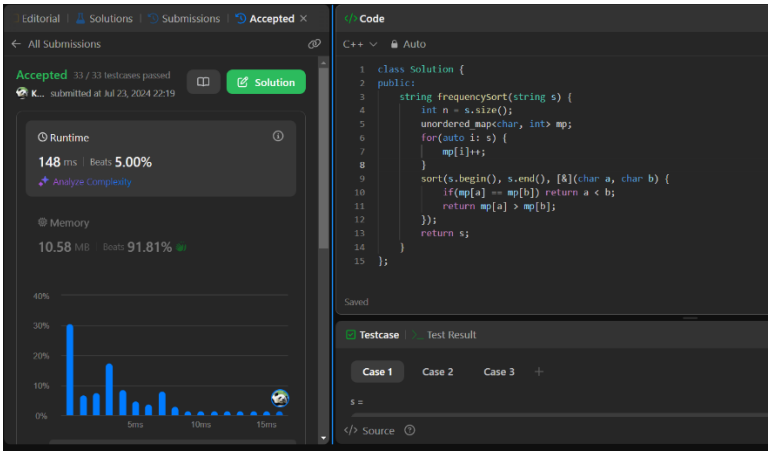
```
class Solution {
public:
    int thirdMax(vector<int>& nums) {
        set<int>st(nums.begin(),nums.end());
        if(st.size() < 3) {
            return *st.rbegin();
        }
        auto i = st.rbegin();
        advance(i,2);
        return *i;
    }
```

};



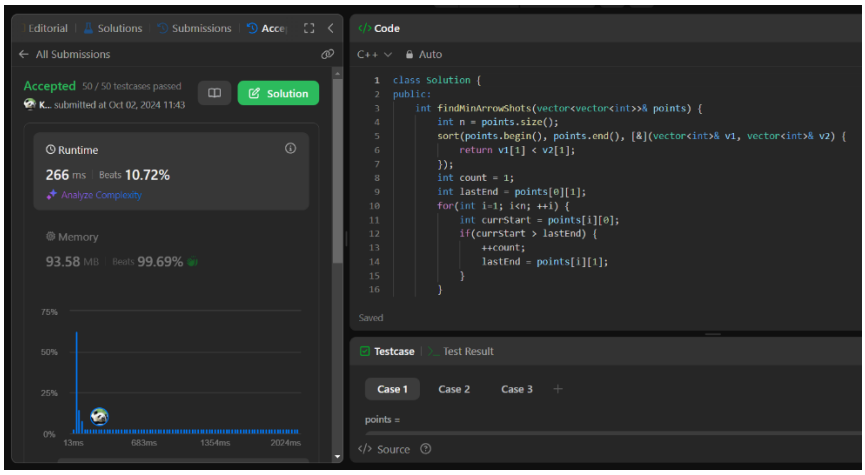
451. Sort Characters By Frequency

```
class Solution {
public:
    string frequencySort(string s) {
        int n = s.size();
        unordered_map<char, int> mp;
        for(auto i: s) {
            mp[i]++;
        }
        sort(s.begin(), s.end(), [&](char a, char b) {
            if(mp[a] == mp[b]) return a < b;
            return mp[a] > mp[b];
        });
        return s;
}
```



452. Minimum Number of Arrows to Burst Balloons

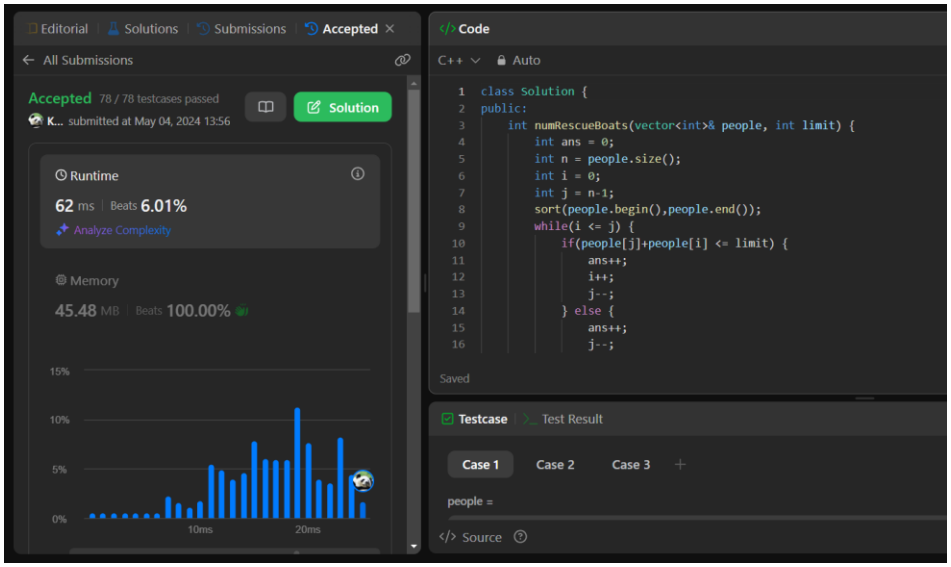
```
class Solution {
public:
   int findMinArrowShots(vector<vector<int>>& points) {
      int n = points.size();
      sort(points.begin(), points.end(), [&](vector<int>& v1, vector<int>& v2) {
         return v1[1] < v2[1];
      });
      int count = 1;
      int lastEnd = points[0][1];
      for(int i=1; i<n; ++i) {
         int currStart = points[i][0];
         if(currStart > lastEnd) {
            ++count;
            lastEnd = points[i][1];
         }
      }
      return count;
   }
};
```



881. Boats to Save People

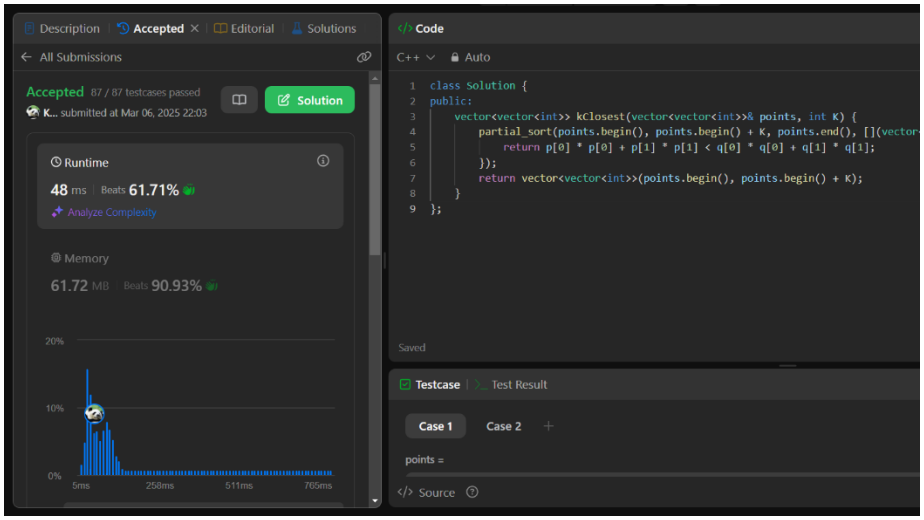
```
class Solution {
public:
   int numRescueBoats(vector<int>& people, int limit) {
      int ans = 0;
      int n = people.size();
      int i = 0;
      int j = n-1;
      sort(people.begin(),people.end());
      while(i <= j) {
         if(people[j]+people[i] <= limit) {
            ans++;
```

```
            i++;
            j--;
        } else {
            ans++;
            j--;
        }
    }
    return ans;
  }
};
```



[973. K Closest Points to Origin]

```
class Solution {
public:
  vector<vector<int>> kClosest(vector<vector<int>>& points, int K) {
    partial_sort(points.begin(), points.begin() + K, points.end(), [](vector<int>& p, vector<int>& q) {
      return p[0] * p[0] + p[1] * p[1] < q[0] * q[0] + q[1] * q[1];
    });
    return vector<vector<int>>(points.begin(), points.begin() + K);
  }
};
```



[1338. Reduce Array Size to The Half]

```
class Solution {
public:
    int minSetSize(vector<int>& arr) {
        int n = arr.size();
        unordered_map<int, int> cnt;
        for (int x : arr) ++cnt[x];
        vector<int> counting(n + 1);
        for (auto [it, freq] : cnt) ++counting[freq];
        int ans = 0, removed = 0, half = n / 2, freq = n;
        while (removed < half) {
            ans += 1;
            while (counting[freq] == 0) --freq;
            removed += freq;
            --counting[freq];
        }
        return ans;
    }
};
```

escription | Editorial | Solutions | Accepted

All Submissions

Accepted 33 / 33 testcases passed

K... submitted at Mar 06, 2025 22:04

Solution

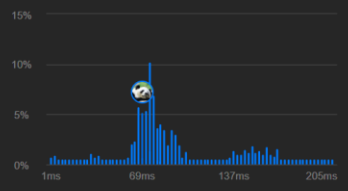
Runtime

71 ms | Beats 79.26%

Analyze Complexity

Memory

81.90 MB | Beats 88.52%



Code

C++ | Auto

```cpp
class Solution {
public:
    int minSetSize(vector<int>& arr) {
        int n = arr.size();
        unordered_map<int, int> cnt;
        for (int x : arr) ++cnt[x];
        vector<int> counting(n + 1);
        for (auto [it, freq] : cnt) ++counting[freq];
        int ans = 0, removed = 0, half = n / 2, freq = n;
        while (removed < half) {
            ans += 1;
            while (counting[freq] == 0) --freq;
            removed += freq;
            --counting[freq];
        }
        return ans;
```

Saved

Testcase | Test Result

Case 1 | Case 2 | +

arr =

Source