AP ASSIGNMENT 5

PRANJAL SINGH

22BCS13041

22BCS_FL_IOT_601_A

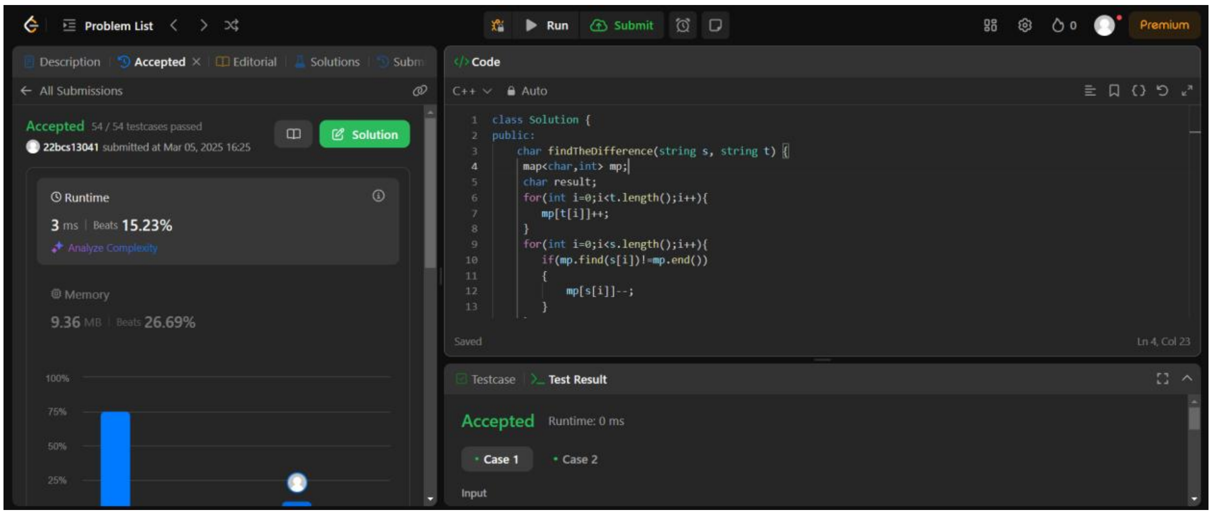
AP ASSIGNMENT 5

Q1. Find the Difference

Implementation Code:

```
class Solution {
public:
    char findTheDifference(string s, string t) {
     map<char,int> mp;
     char result;
     for(int i=0;i<t.length();i++){
        mp[t[i]]++;
     }
     for(int i=0;i<s.length();i++){
        if(mp.find(s[i])!=mp.end())
        {
            mp[s[i]]--;
        }
     }
     for(auto it : mp)
        {
            if(it.second>=1)
            result=it.first;
        }
     return result;
    }
};
```

Output:



Accepted 54 / 54 testcases passed
22bcs13041 submitted at Mar 05, 2025 16:25

Runtime
3 ms | Beats 15.23%
Analyze Complexity

Memory
9.36 MB | Beats 26.69%

```
class Solution {
public:
    char findTheDifference(string s, string t) {
     map<char,int> mp;
     char result;
     for(int i=0;i<t.length();i++){
        mp[t[i]]++;
     }
     for(int i=0;i<s.length();i++){
        if(mp.find(s[i])!=mp.end())
        {
            mp[s[i]]--;
        }
```

Saved   Ln 4, Col 23

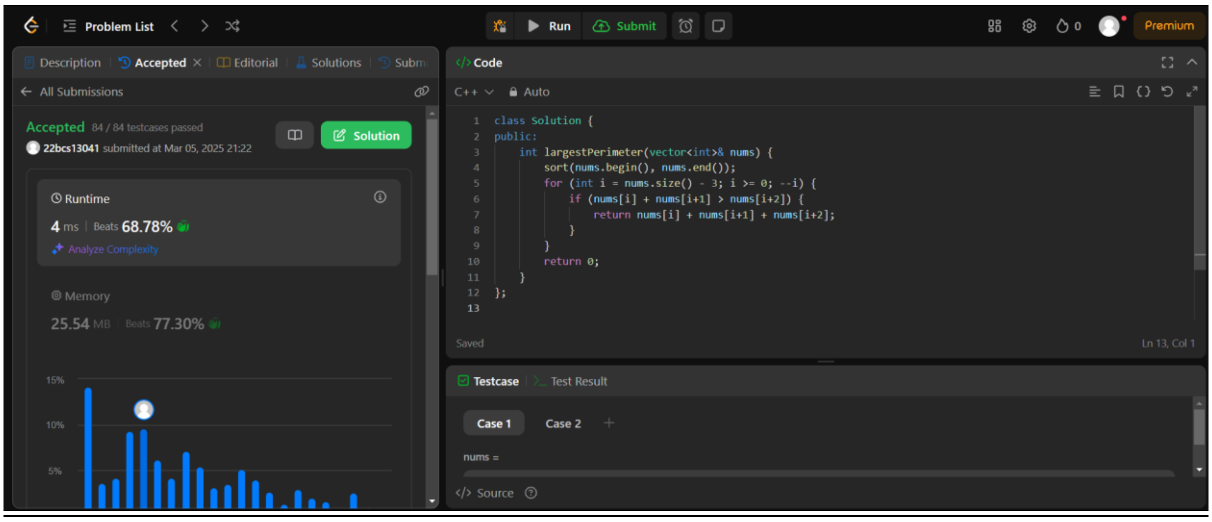
Accepted Runtime: 0 ms

Q2. Largest Perimeter Triangle

Implementation Code:

```
class Solution {
public:
    int largestPerimeter(vector<int>& nums) {
        sort(nums.begin(), nums.end());
        for (int i = nums.size() - 3; i >= 0; --i) {
            if (nums[i] + nums[i+1] > nums[i+2]) {
                return nums[i] + nums[i+1] + nums[i+2];
            }
        }
        return 0;
    }
};
```

Output:



Accepted 84 / 84 testcases passed
22bcs13041 submitted at Mar 05, 2025 21:22

Runtime
4 ms | Beats 68.78%

Memory
25.54 MB | Beats 77.30%

```
class Solution {
public:
    int largestPerimeter(vector<int>& nums) {
        sort(nums.begin(), nums.end());
        for (int i = nums.size() - 3; i >= 0; --i) {
            if (nums[i] + nums[i+1] > nums[i+2]) {
                return nums[i] + nums[i+1] + nums[i+2];
            }
        }
        return 0;
    }
};
```

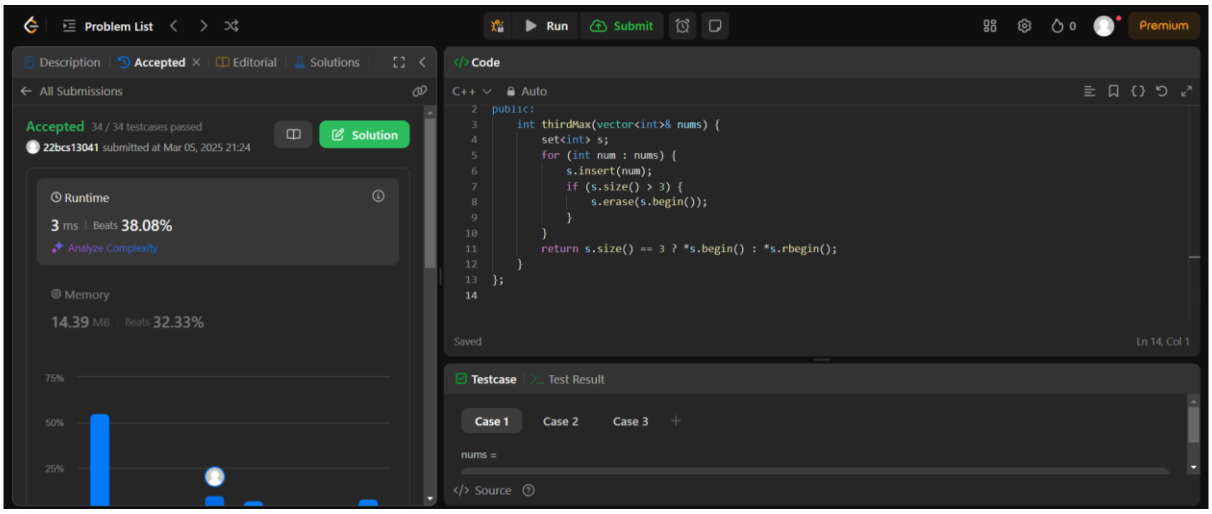
Q3. Third Maximum Number

Implementation Code:

```
class Solution {
public:
    int thirdMax(vector<int>& nums) {
        set<int> s;
```

```cpp
        for (int num : nums) {
            s.insert(num);
            if (s.size() > 3) {
                s.erase(s.begin());
            }
        }
        return s.size() == 3 ? *s.begin() : *s.rbegin();
    }
};
```

<u>Output:</u>



Accepted 34 / 34 testcases passed
22bcs13041 submitted at Mar 05, 2025 21:24

Runtime
3 ms | Beats 38.08%
Analyze Complexity

Memory
14.39 MB | Beats 32.33%

```cpp
public:
    int thirdMax(vector<int>& nums) {
        set<int> s;
        for (int num : nums) {
            s.insert(num);
            if (s.size() > 3) {
                s.erase(s.begin());
            }
        }
        return s.size() == 3 ? *s.begin() : *s.rbegin();
    }
};
```

Saved
Ln 14, Col 1

nums =

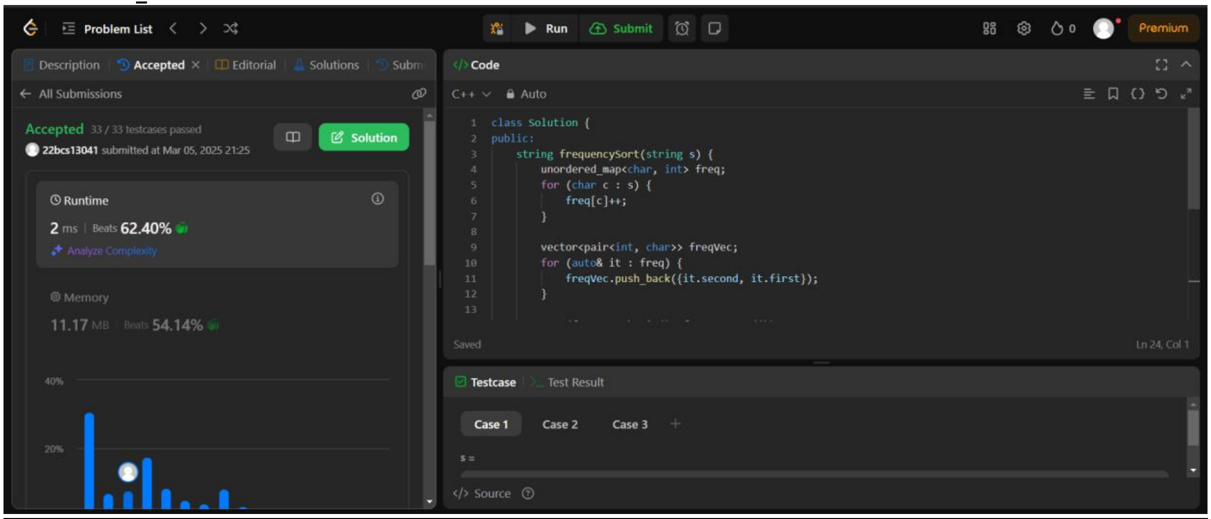
Q4. Sort Characters by frequency

<u>Implementation Code:</u>

```cpp
class Solution {
public:
    string frequencySort(string s) {
        unordered_map<char, int> freq;
        for (char c : s) {
            freq[c]++;
        }

        vector<pair<int, char>> freqVec;
        for (auto& it : freq) {
            freqVec.push_back({it.second, it.first});
        }

        sort(freqVec.rbegin(), freqVec.rend());
        string result;
        for (auto& p : freqVec) {
            result.append(p.first, p.second);
        }

        return result;
    }
};
```

Output

:



Accepted 33 / 33 testcases passed
22bcs13041 submitted at Mar 05, 2025 21:25

Runtime
2 ms | Beats 62.40%

Memory
11.17 MB | Beats 54.14%

```
class Solution {
public:
    string frequencySort(string s) {
        unordered_map<char, int> freq;
        for (char c : s) {
            freq[c]++;
        }

        vector<pair<int, char>> freqVec;
        for (auto& it : freq) {
            freqVec.push_back({it.second, it.first});
        }

```

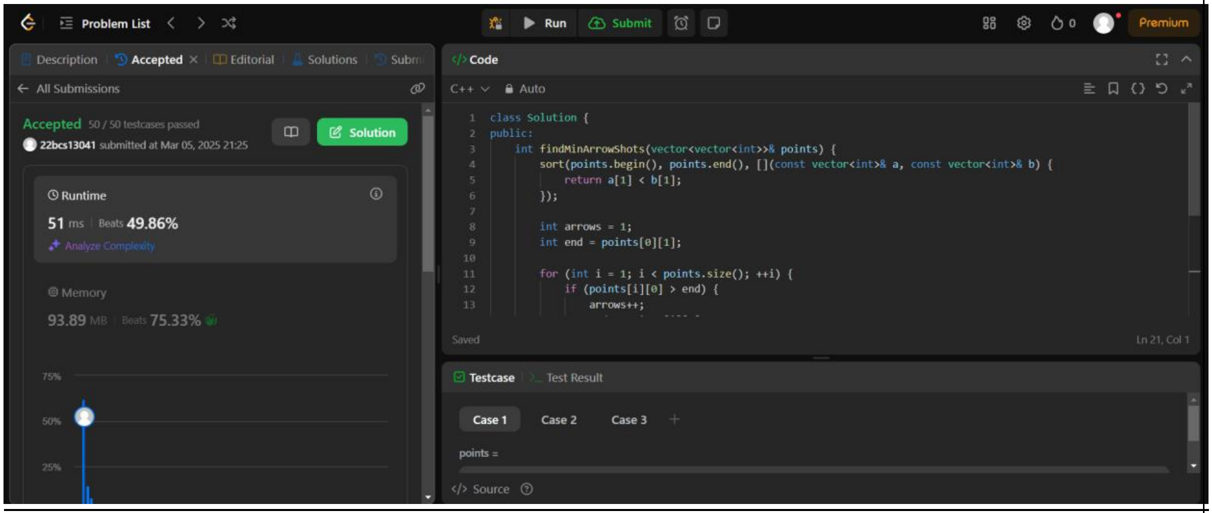
Saved   Ln 24, Col 1

Q5. Minimum Number of Arrows to Burst Balloons

Implementation Code:

```
class Solution {
public:
    int findMinArrowShots(vector<vector<int>>& points) {
        sort(points.begin(), points.end(), [](const vector<int>& a, const vector<int>& b) {
            return a[1] < b[1];
        });

        int arrows = 1;
        int end = points[0][1];

        for (int i = 1; i < points.size(); ++i) {
            if (points[i][0] > end) {
                arrows++;
                end = points[i][1];
            }
        }
        return arrows;
    }
};
```

Output:



Accepted 50 / 50 testcases passed
22bcs13041 submitted at Mar 05, 2025 21:25

Runtime
51 ms | Beats 49.86%

Memory
93.89 MB | Beats 75.33%

```
class Solution {
public:
    int findMinArrowShots(vector<vector<int>>& points) {
        sort(points.begin(), points.end(), [](const vector<int>& a, const vector<int>& b) {
            return a[1] < b[1];
        });

        int arrows = 1;
        int end = points[0][1];

        for (int i = 1; i < points.size(); ++i) {
            if (points[i][0] > end) {
                arrows++;
```

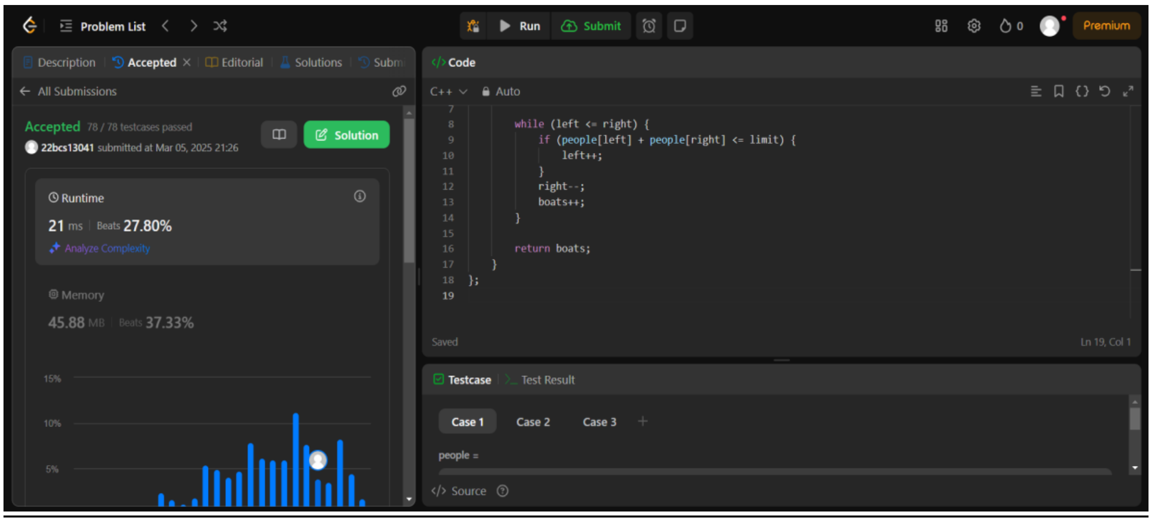
Saved   Ln 21, Col 1

Q6. Boats to save people

Implementation Code:

```
class Solution {
public:
    int numRescueBoats(vector<int>& people, int limit) {
        sort(people.begin(), people.end());
        int left = 0, right = people.size() - 1;
        int boats = 0;

        while (left <= right) {
            if (people[left] + people[right] <= limit) {
                left++;
            }
            right--;
            boats++;
        }

        return boats;
    }
};
```

Output:



Accepted 78 / 78 testcases passed

22bcs13041 submitted at Mar 05, 2025 21:26

Runtime

21 ms | Beats 27.80%

Analyze Complexity

Memory

45.88 MB | Beats 37.33%

```
        while (left <= right) {
            if (people[left] + people[right] <= limit) {
                left++;
            }
            right--;
            boats++;
        }

        return boats;
    }
};
```

Saved

Ln 19, Col 1

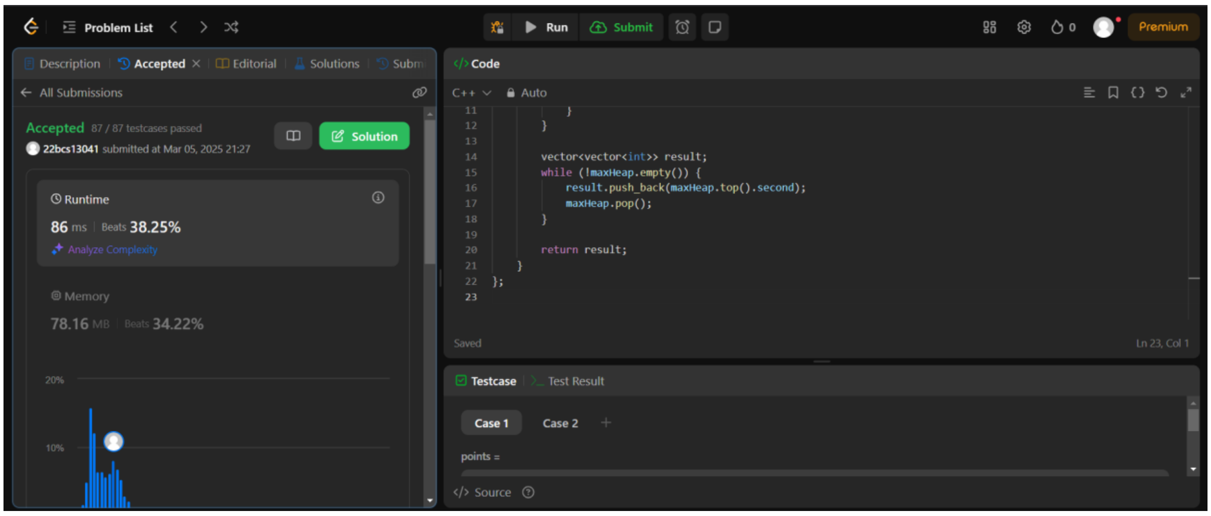
people =

Q7. K closest points to origin

Implementation Code:

```
class Solution {
public:
    vector<vector<int>> kClosest(vector<vector<int>>& points, int k) {
        priority_queue<pair<int, vector<int>>> maxHeap;

        for (auto& point : points) {
            int dist = point[0] * point[0] + point[1] * point[1];
            maxHeap.push({dist, point});
            if (maxHeap.size() > k) {
                maxHeap.pop();
            }
        }

        vector<vector<int>> result;
        while (!maxHeap.empty()) {
            result.push_back(maxHeap.top().second);
            maxHeap.pop();
        }

        return result;
    }
};
```

Output:



Accepted 87 / 87 testcases passed
22bcs13041 submitted at Mar 05, 2025 21:27

Runtime
86 ms | Beats 38.25%

Memory
78.16 MB | Beats 34.22%

```
            }
        }

        vector<vector<int>> result;
        while (!maxHeap.empty()) {
            result.push_back(maxHeap.top().second);
            maxHeap.pop();
        }

        return result;
    }
};
```

Saved   Ln 23, Col 1

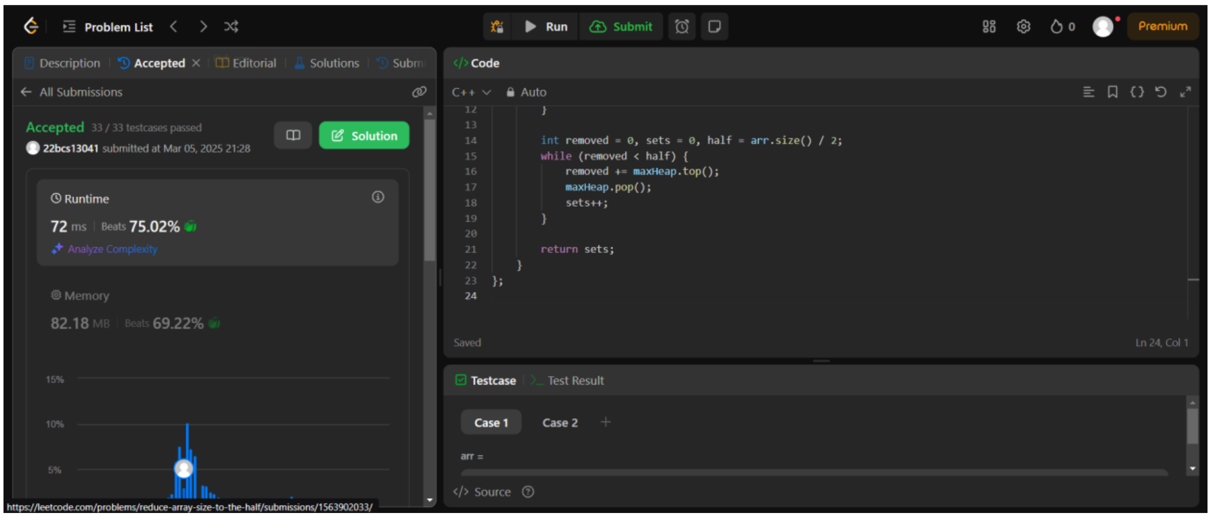
points =

Q8. Reduce array size to half

Implementation Code:

```
class Solution {
public:
    int minSetSize(vector<int>& arr) {
        unordered_map<int, int> freq;
        for (int num : arr) {
            freq[num]++;
        }

        priority_queue<int> maxHeap;
        for (auto& [key, count] : freq) {
            maxHeap.push(count);
        }

        int removed = 0, sets = 0, half = arr.size() / 2;
        while (removed < half) {
            removed += maxHeap.top();
            maxHeap.pop();
            sets++;
        }

        return sets;
    }
};
```

Output:



Accepted 33 / 33 testcases passed

22bcs13041 submitted at Mar 05, 2025 21:28

Runtime

72 ms | Beats 75.02%

Memory

82.18 MB | Beats 69.22%

```
        }

        int removed = 0, sets = 0, half = arr.size() / 2;
        while (removed < half) {
            removed += maxHeap.top();
            maxHeap.pop();
            sets++;
        }

        return sets;
    }
};
```

https://leetcode.com/problems/reduce-array-size-to-the-half/submissions/1563902033/