Name: Anish Patial

UID: 22BCS15029

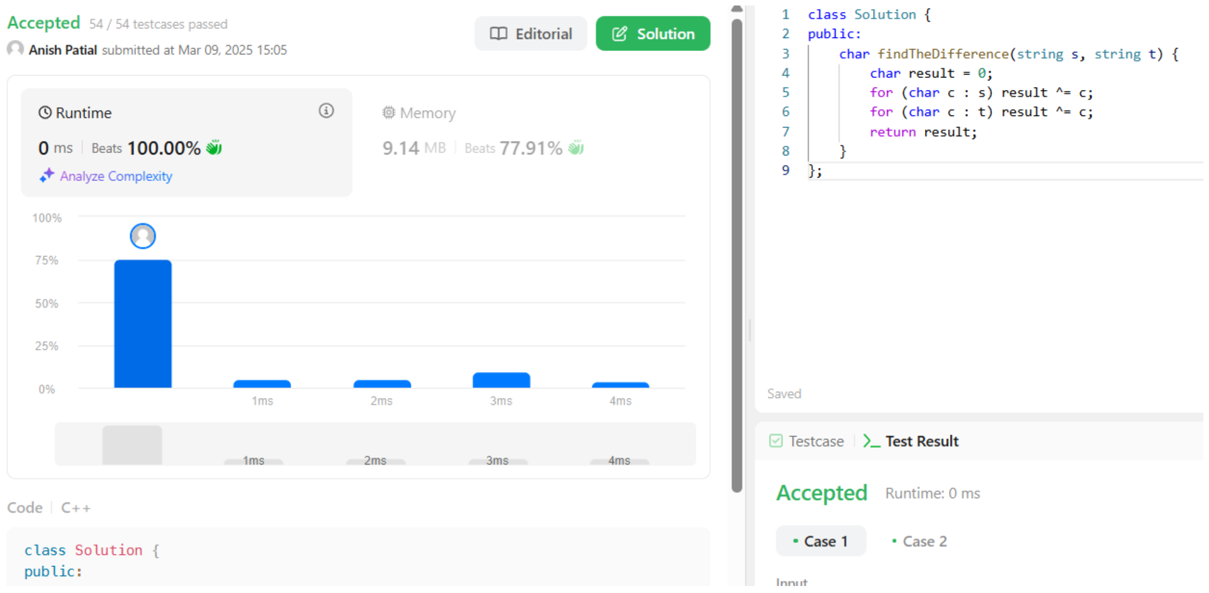
Section: FL_IOT_601 - A

Assignment – 5 Solutions:-

1. Find the Difference:-

```
class Solution {
public:
    char findTheDifference(string s, string t) {
        char result = 0;
        for (char c : s) result ^= c;
        for (char c : t) result ^= c;
        return result;
    }
};
```

Result:-



Accepted 54 / 54 testcases passed
Anish Patial submitted at Mar 09, 2025 15:05

Runtime: 0 ms | Beats 100.00%
Memory: 9.14 MB | Beats 77.91%

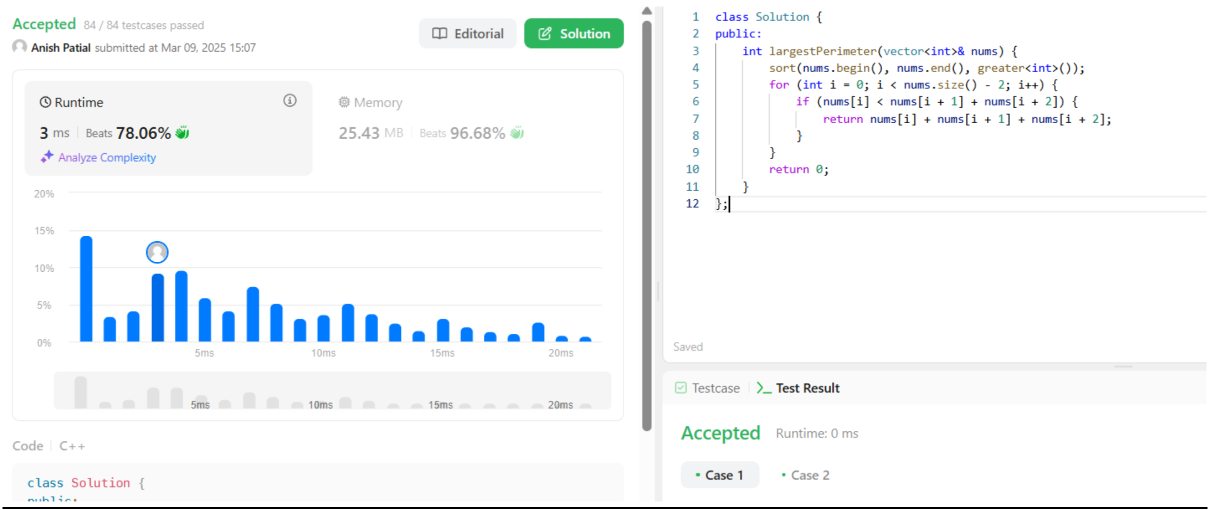
```
class Solution {
public:
    char findTheDifference(string s, string t) {
        char result = 0;
        for (char c : s) result ^= c;
        for (char c : t) result ^= c;
        return result;
    }
};
```

Accepted Runtime: 0 ms

2. Largest Perimeter Triangle:

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

Result:



Accepted 84 / 84 testcases passed
Anish Patial submitted at Mar 09, 2025 15:07

Runtime
3 ms | Beats 78.06%

Memory
25.43 MB | Beats 96.68%

Code | C++

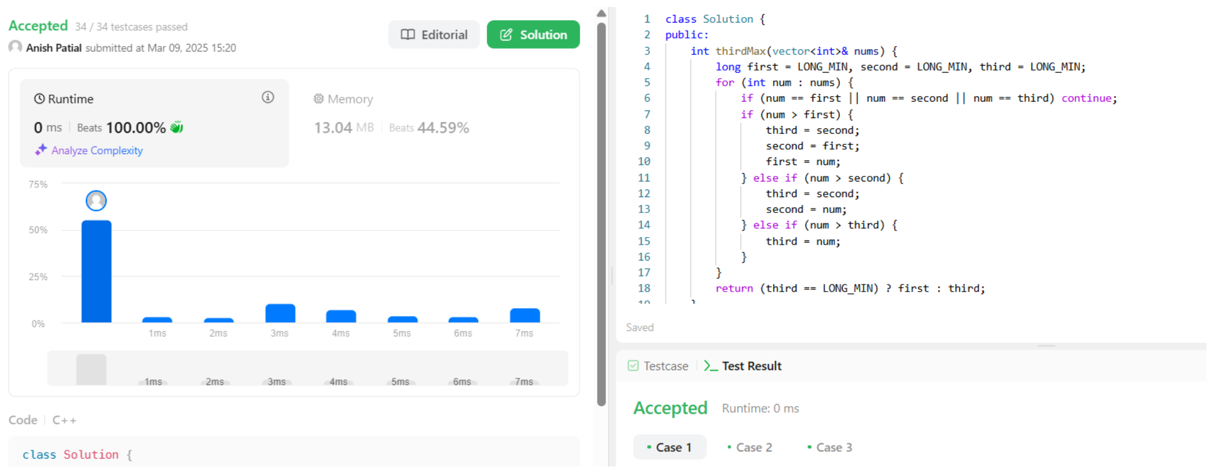
```
class Solution {
public:
    int largestPerimeter(vector<int>& nums) {
        sort(nums.begin(), nums.end(), greater<int>());
        for (int i = 0; i < nums.size() - 2; i++) {
            if (nums[i] < nums[i + 1] + nums[i + 2]) {
                return nums[i] + nums[i + 1] + nums[i + 2];
  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

Accepted Runtime: 0 ms

3. **Third Maximum Number:**

```
class Solution {
public:
    int thirdMax(vector<int>& nums) {
        long first = LONG_MIN, second = LONG_MIN, third = LONG_MIN;
        for (int num : nums) {
            if (num == first || num == second || num == third) continue;
            if (num > first) {
                third = second;
                second = first;
                first = num;
            } else if (num > second) {
                third = second;
                second = num;
            } else if (num > third) {
                third = num;
            }
        }
        return (third == LONG_MIN) ? first : third;
    }
};
```

<u>Result:</u>



Accepted 34 / 34 testcases passed

Anish Patial submitted at Mar 09, 2025 15:20

```
class Solution {
```

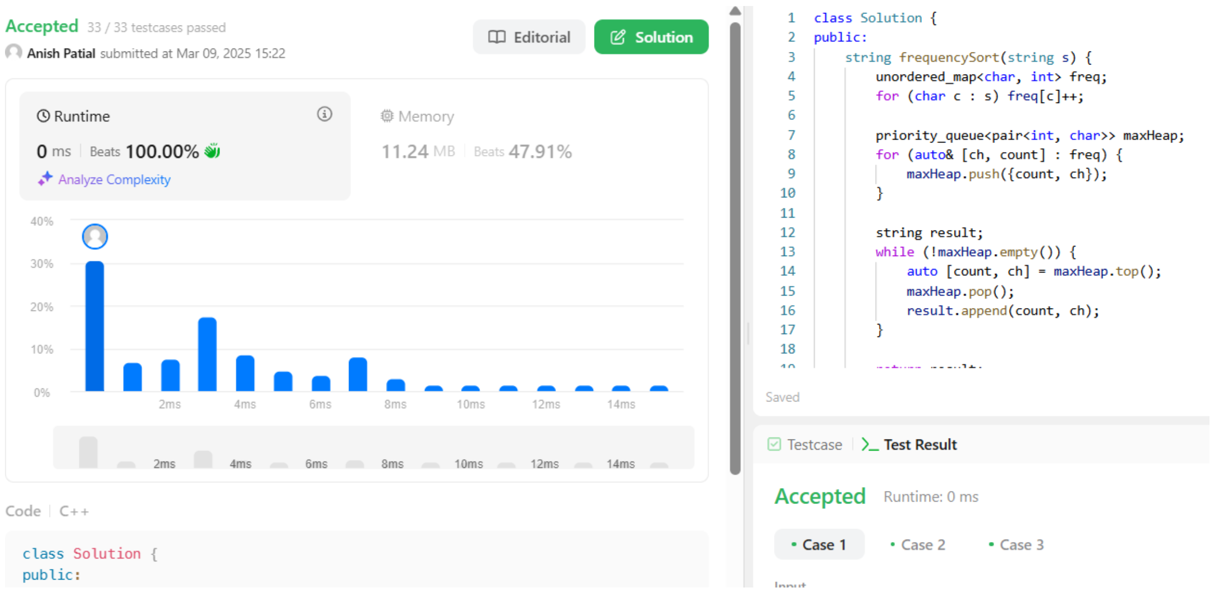
```
class Solution {
public:
    int thirdMax(vector<int>& nums) {
        long first = LONG_MIN, second = LONG_MIN, third = LONG_MIN;
        for (int num : nums) {
            if (num == first || num == second || num == third) continue;
            if (num > first) {
                third = second;
                second = first;
                first = num;
            } else if (num > second) {
                third = second;
                second = num;
            } else if (num > third) {
                third = num;
            }
        }
        return (third == LONG_MIN) ? first : third;
```

Accepted Runtime: 0 ms

4. <u>Sort Characters By Frequency:</u>

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

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

        string result;
        while (!maxHeap.empty()) {
            auto [count, ch] = maxHeap.top();
            maxHeap.pop();
            result.append(count, ch);
        }
        return result;
    }
};
```

Result:



Accepted 33 / 33 testcases passed

Anish Patial submitted at Mar 09, 2025 15:22

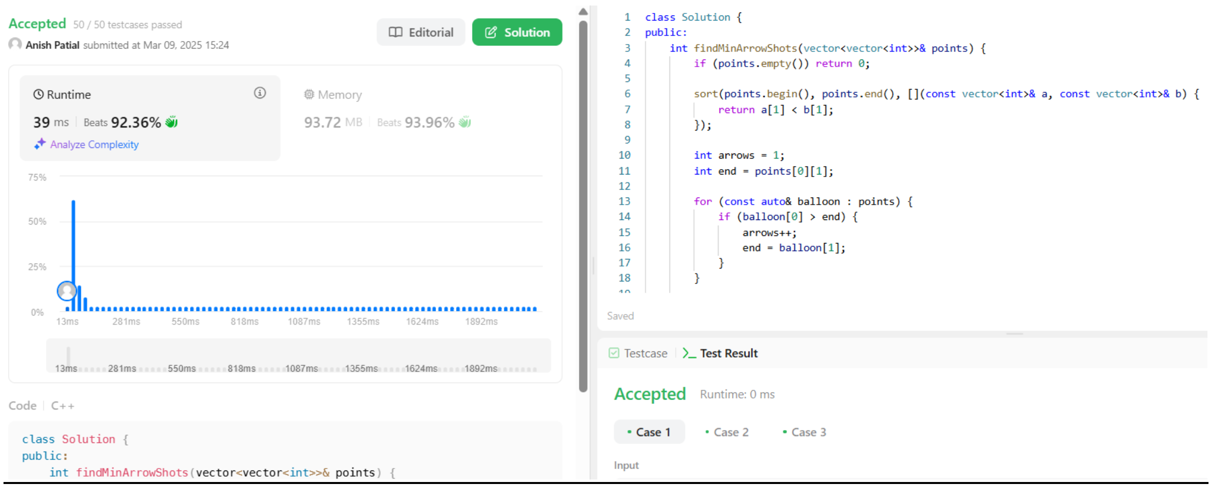
```
class Solution {
public:
    string frequencySort(string s) {
        unordered_map<char, int> freq;
        for (char c : s) freq[c]++;

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

        string result;
        while (!maxHeap.empty()) {
            auto [count, ch] = maxHeap.top();
            maxHeap.pop();
            result.append(count, ch);
        }

```

Accepted Runtime: 0 ms

5. Minimum Number of Arrows to Burst Balloons:

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

        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 (const auto& balloon : points) {
            if (balloon[0] > end) {
                arrows++;
                end = balloon[1];
            }
        }

        return arrows;
    }
};
```

Result:



Accepted 50 / 50 testcases passed
Anish Patial submitted at Mar 09, 2025 15:24

Runtime
39 ms | Beats 92.36%
Analyze Complexity

Memory
93.72 MB | Beats 93.96%

Code | C++

```
class Solution {
public:
    int findMinArrowShots(vector<vector<int>>& points) {
```

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

        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 (const auto& balloon : points) {
            if (balloon[0] > end) {
                arrows++;
                end = balloon[1];
            }
        }
```

Saved

Accepted Runtime: 0 ms

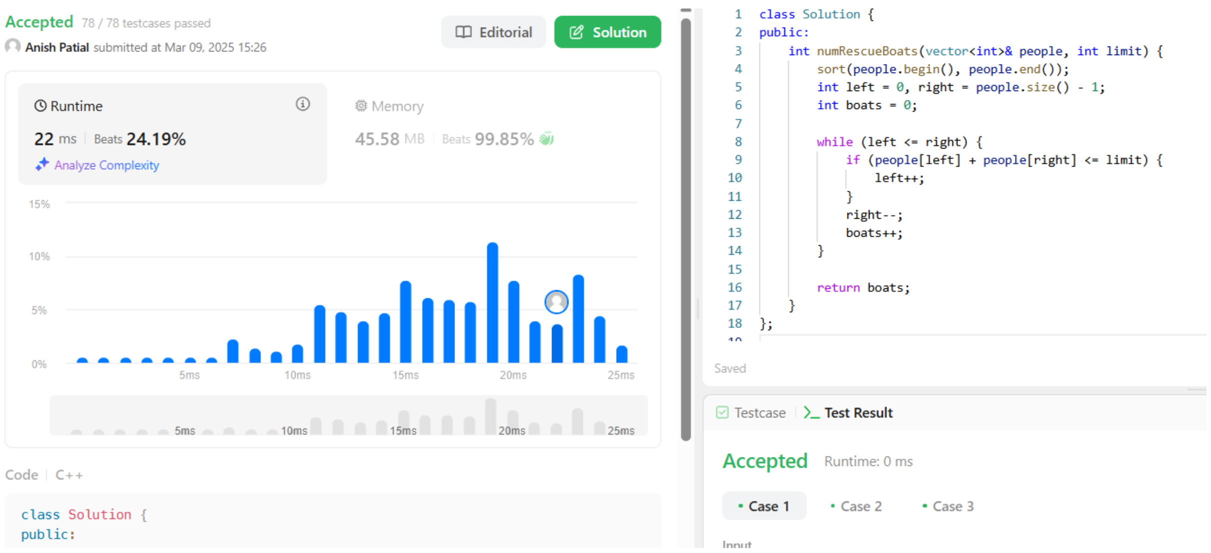
• Case 1 • Case 2 • Case 3

Input

6. **Boats to Save People:**

```
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;
    }
};
```

Result:



Accepted 78 / 78 testcases passed

Anish Patial submitted at Mar 09, 2025 15:26

Runtime

22 ms | Beats 24.19%

Memory

45.58 MB | Beats 99.85%

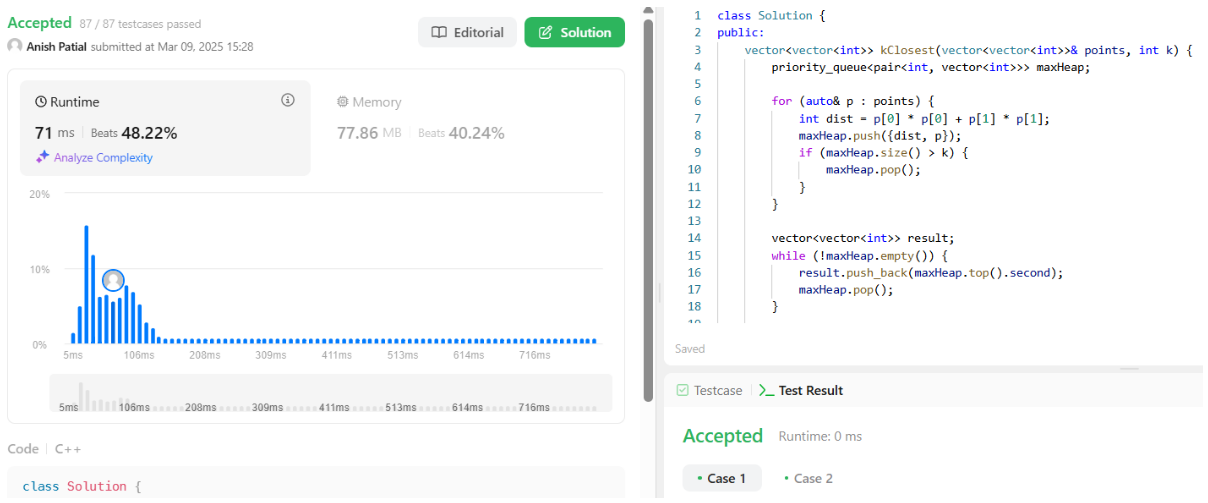
```
class Solution {
public:
    int numRescueBoats(vector<int>& people, int limit) {
        sort(people.begin(), people.end());
        int left = 0, right = people.size() - 1;
        int boats = 0;

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

        return boats;
    }
};
```

Accepted Runtime: 0 ms

7. K Closest Points to Origin:

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

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

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

        return result;
    }
};
```

Result:



Accepted 87 / 87 testcases passed

Anish Patial submitted at Mar 09, 2025 15:28

Runtime 71 ms Beats 48.22%

Memory 77.86 MB Beats 40.24%

Code | C++

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

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

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

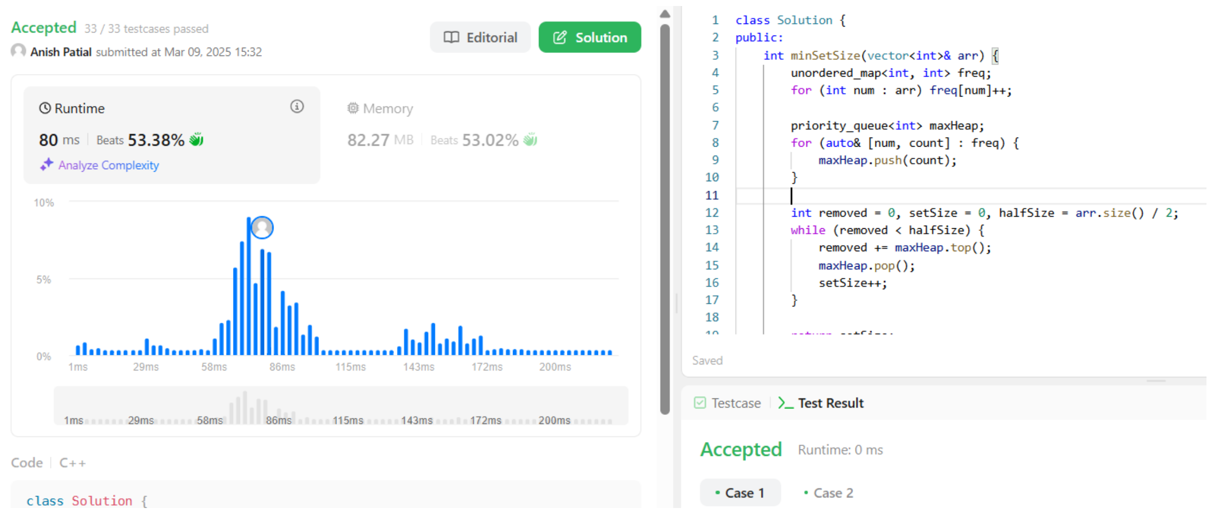
Saved

Accepted Runtime: 0 ms

8. **Reduce Array Size to The Half:**

```
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& [num, count] : freq) {
            maxHeap.push(count);
        }

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

        return setSize;
    }
};
```

Result:-



Accepted 33 / 33 testcases passed

Anish Patial submitted at Mar 09, 2025 15:32

Runtime: 80 ms | Beats 53.38%

Memory: 82.27 MB | Beats 53.02%

Code | C++

```
class Solution {
```

```
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& [num, count] : freq) {
            maxHeap.push(count);
        }

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

```

Saved

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 · Case 2