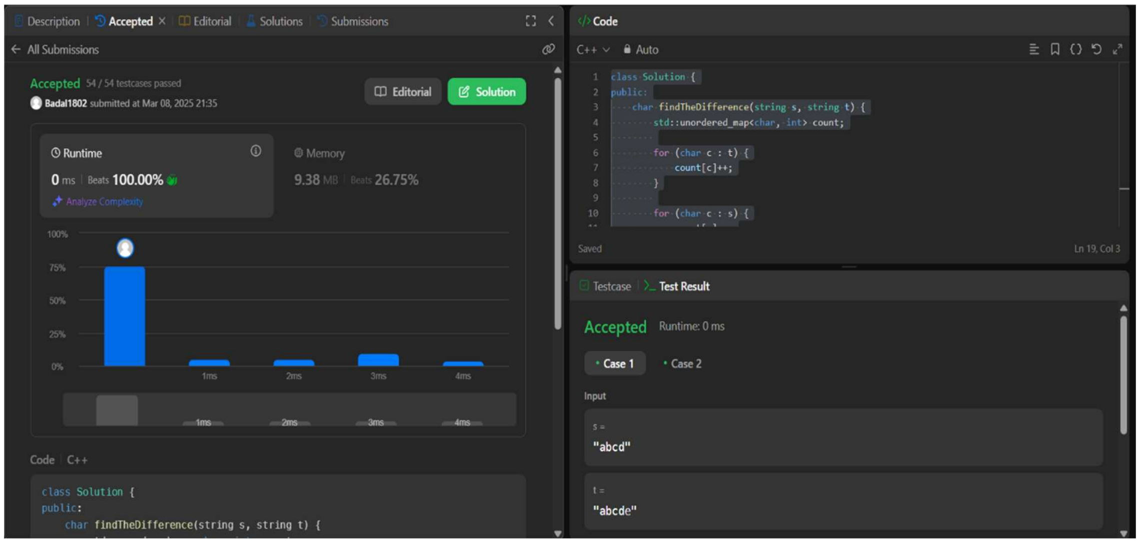
389.Find the diffrence

CODE:

```
class Solution {
public:
  char findTheDifference(string s, string t) {
    std::unordered_map<char, int> count;
        for (char c : t) {
      count[c]++;
    }
      for (char c : s) {
      count[c]--;
      if (count[c] == 0) {
        count.erase(c);
      }
    }
    return count.begin()->first;
  }
};
```

OUTPUT:



Accepted 54 / 54 testcases passed

Badal1802 submitted at Mar 08, 2025 21:35

Runtime

0 ms | Beats 100.00%

Memory

9.38 MB | Beats 26.75%

Code | C++

```
class Solution {
public:
    char findTheDifference(string s, string t) {
```

```
class Solution {
public:
    char findTheDifference(string s, string t) {
        std::unordered_map<char, int> count;

        for (char c : t) {
            count[c]++;
        }

        for (char c : s) {
```

Saved

Ln 19, Col 3

Accepted Runtime: 0 ms

Input

s =

"abcd"

t =

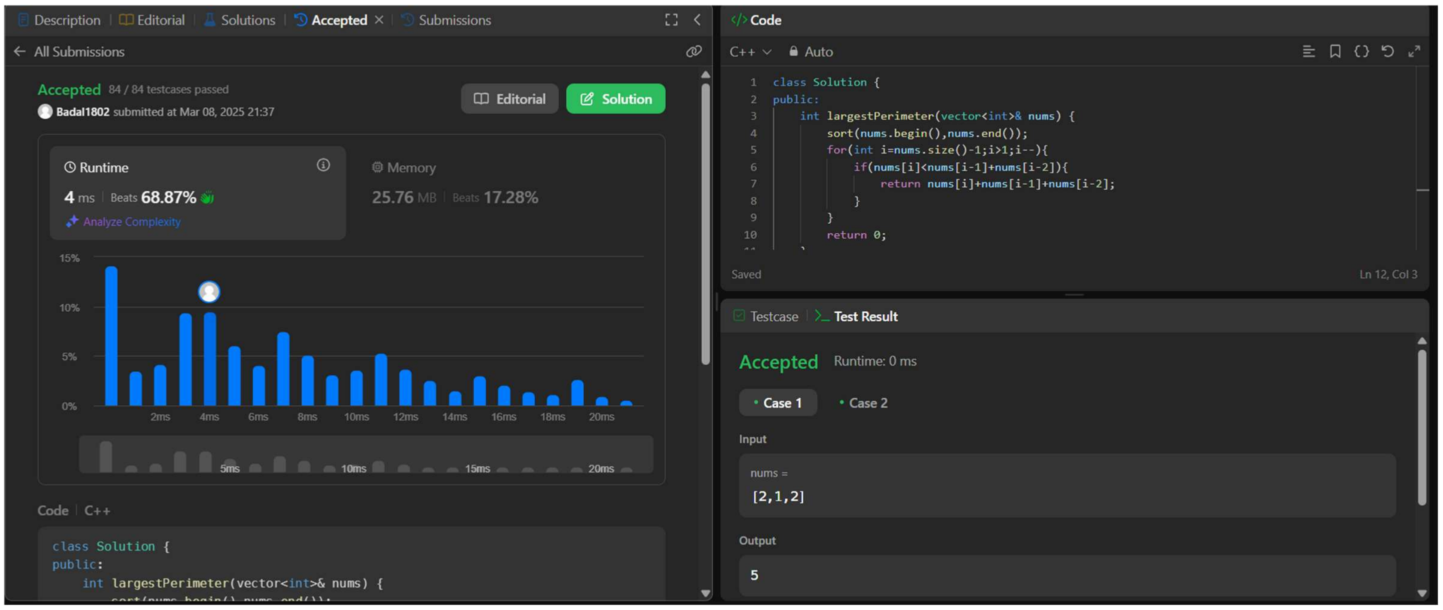
"abcde"

976.Largest Perimeter Triangle

CODE:

```
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;
    }
};
```

OUTPUT:



Accepted 84 / 84 testcases passed

Badal1802 submitted at Mar 08, 2025 21:37

Runtime 4 ms | Beats 68.87%

Memory 25.76 MB | Beats 17.28%

Code | C++

```
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;
```

Accepted Runtime: 0 ms

Input

```
nums =
[2,1,2]
```

Output

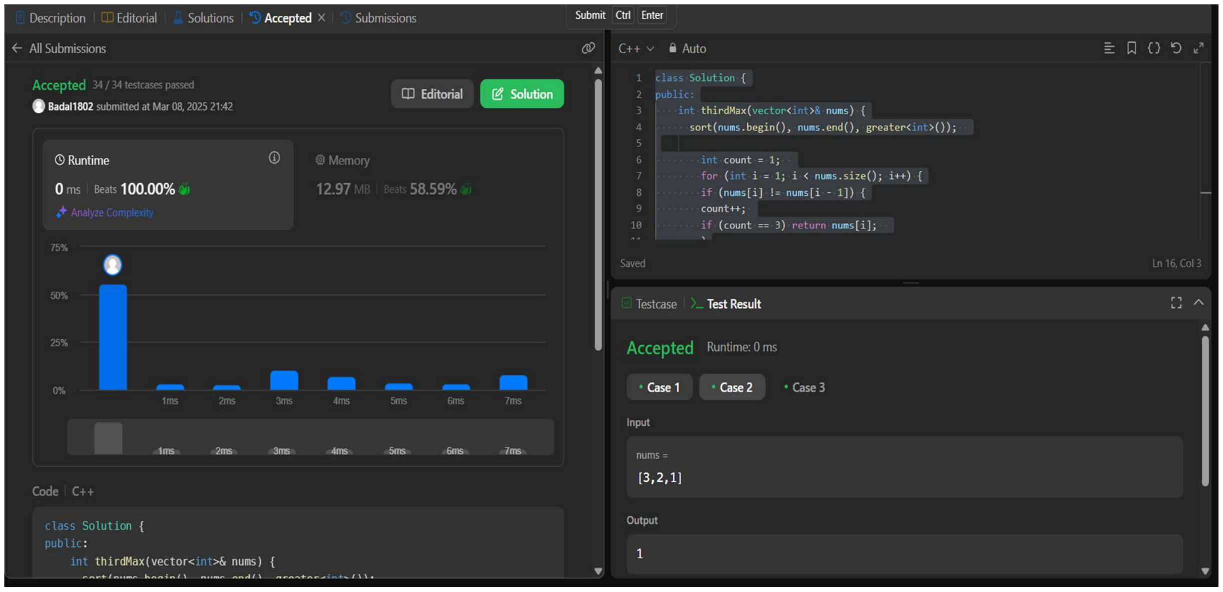
```
5
```

414.Third Maximum Number

CODE:

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

OUTPUT:



Accepted 34 / 34 testcases passed

Badal1802 submitted at Mar 08, 2025 21:42

Runtime 0 ms | Beats 100.00%

Memory 12.97 MB | Beats 58.59%

Code | C++

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

```
class Solution {
public:
    int thirdMax(vector<int>& nums) {
      sort(nums.begin(), nums.end(), greater<int>());

        int count = 1;
        for (int i = 1; i < nums.size(); i++) {
        if (nums[i] != nums[i - 1]) {
        count++;
        if (count == 3) return nums[i];
```

Saved   Ln 16, Col 3

Accepted Runtime: 0 ms

Input

```
nums =
[3,2,1]
```

Output

```
1
```

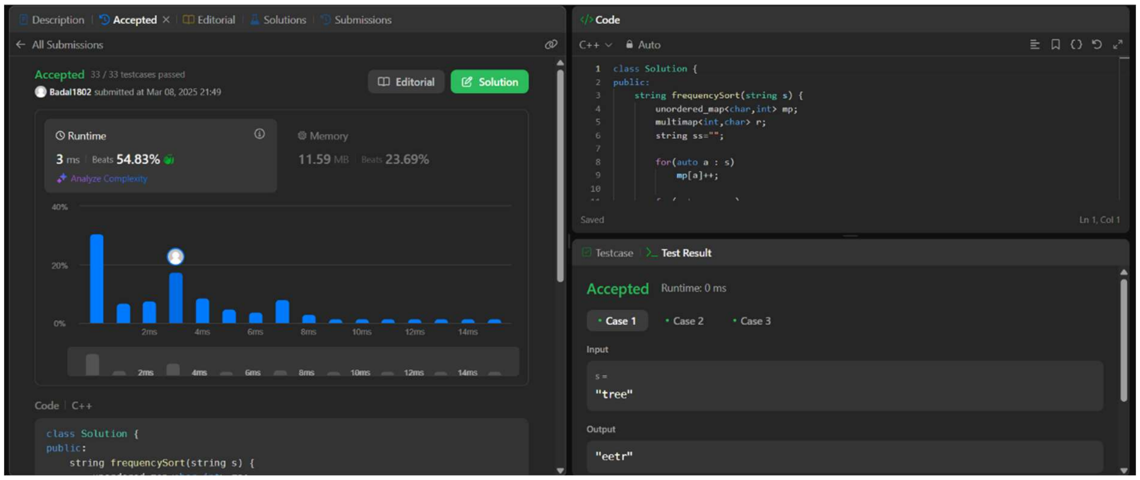
};

451.Sort Characters By Frequency

CODE:

```
class Solution {
public:
    string frequencySort(string s) {

        unordered_map<char,int> mp;

        multimap<int,char> r;
        string ss="";

        for(auto a : s)
            mp[a]++;

        for(auto a : mp)
            r.insert({a.second, a.first});

        for(auto it = r.rbegin(); it != r.rend(); ++it)

            ss += string(it->first, it->second);

        //for(auto it = r.rbegin(); it != r.rend(); ++it){
        //   for (int i = 0; i < it->first; ++i) {
        //         ss += it->second;
        //       }
        //}

        return ss;
    }
};
```

OUTPUT:


Accepted 33 / 33 testcases passed
Badal1802 submitted at Mar 08, 2025 21:49

Runtime
3 ms | Beats 54.83%

Memory
11.59 MB | Beats 23.69%

Code | C++

```
class Solution {
public:
    string frequencySort(string s) {
```

```
class Solution {
public:
    string frequencySort(string s) {
        unordered_map<char,int> mp;
        multimap<int,char> r;
        string ss="";

        for(auto a : s)
            mp[a]++;
```

Saved   Ln 1, Col 1

Accepted   Runtime: 0 ms

Input
s =
"tree"

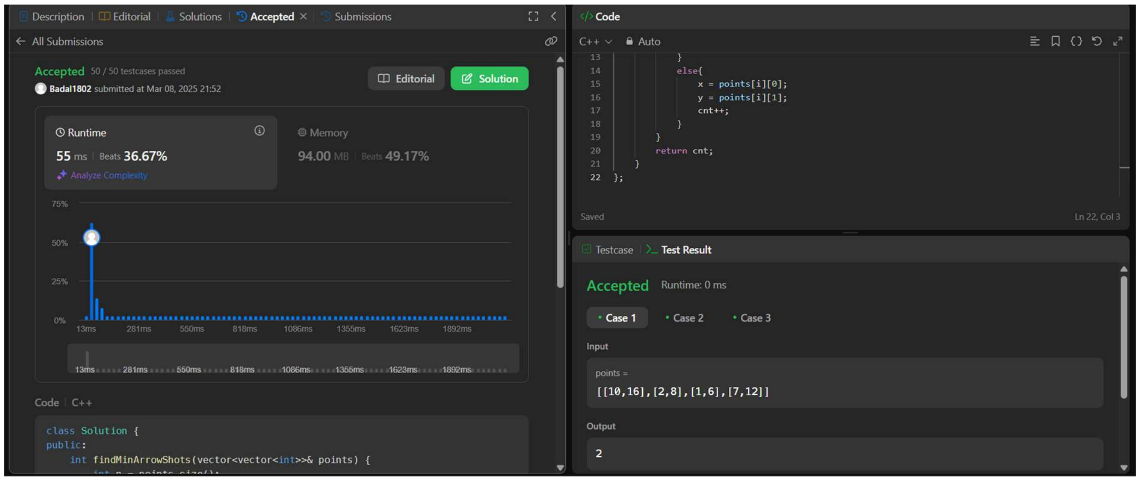
Output
"eetr"

452.Minimum Number of Arrows to Burst Balloons

CODE:

```
Class Solution {
public:
    int findMinArrowShots(vector<vector<int>>& points) {
        int n = points.size();
        sort(points.begin(), points.end());
        int cnt = 1;
        int x = points[0][0], y = points[0][1];
        for(int i = 1; i < n; i++){
            if((points[i][0] >= x && points[i][0] <= y) ||
               (points[i][1] >= x && points[i][1] <= y)){
                x = max(x, points[i][0]);
                y = min(y, points[i][1]);
            }
            else{
                x = points[i][0];
                y = points[i][1];
                cnt++;
            }
        }
        return cnt;
    }
};
```

OUTPUT:



Accepted 50 / 50 testcases passed

Badal1802 submitted at Mar 08, 2025 21:52

Runtime
55 ms | Beats 36.67%

Memory
94.00 MB | Beats 49.17%

Code | C++

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

```
            }
            else{
                x = points[i][0];
                y = points[i][1];
                cnt++;
            }
        }
        return cnt;
    }
};
```

Accepted Runtime: 0 ms

Input

```
points =
[[10,16],[2,8],[1,6],[7,12]]
```

Output

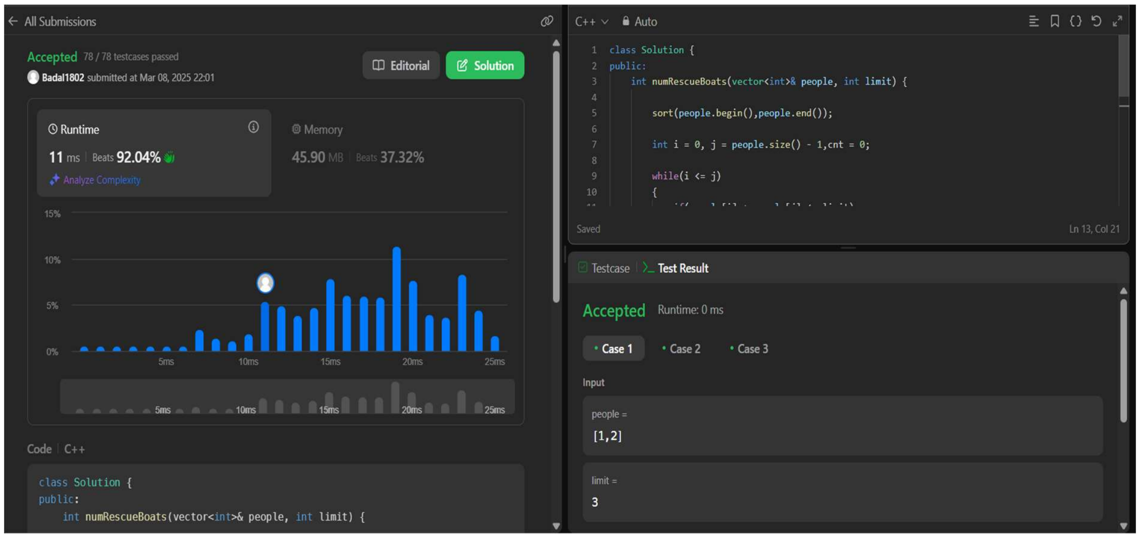
```
2
```

881.Boats to Save People

CODE:

```
class Solution {
public:
    int numRescueBoats(vector<int>& people, int limit) {
        sort(people.begin(), people.end());

        int i = 0, j = people.size() - 1, cnt = 0;

        while (i <= j) {
            if (people[i] + people[j] <= limit) {
                ++i;
                --j;
            } else {
                --j;
            }
            ++cnt;
        }
        return cnt;
    }
};
```

OUTPUT:



All Submissions

Accepted 78 / 78 testcases passed
Badal1802 submitted at Mar 08, 2025 22:01

Runtime
11 ms | Beats 92.04%

Memory
45.90 MB | Beats 37.32%

Code | C++

```
class Solution {
public:
    int numRescueBoats(vector<int>& people, int limit) {
```

C++ Auto

```
class Solution {
public:
    int numRescueBoats(vector<int>& people, int limit) {

        sort(people.begin(),people.end());

        int i = 0, j = people.size() - 1,cnt = 0;

        while(i <= j)
        {
```

Saved Ln 13, Col 21

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

people =
[1,2]

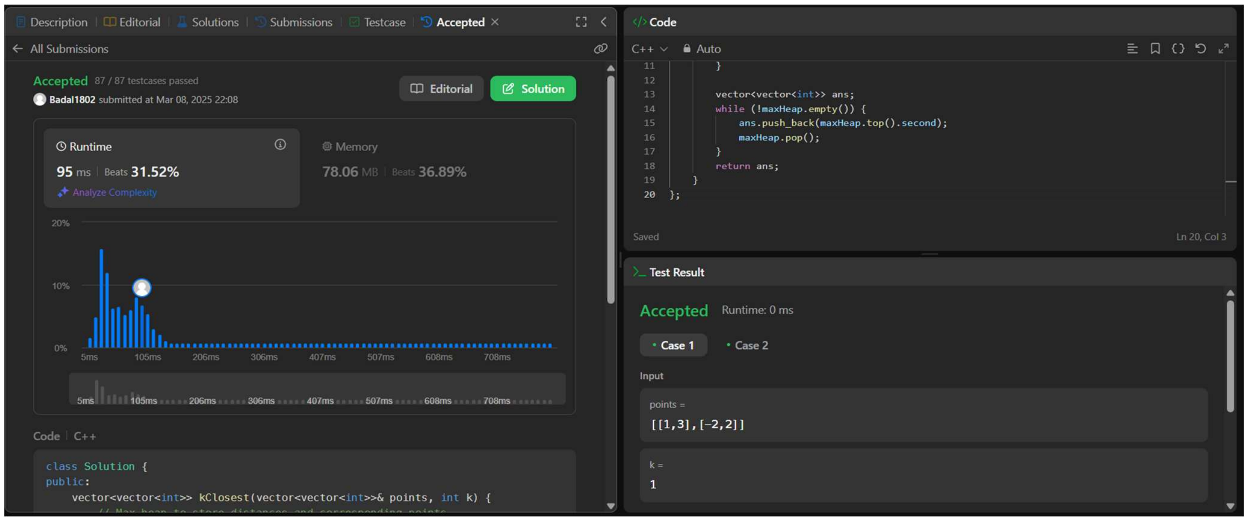
limit =
3

973.K Closest Points to Origin

CODE:

```
class Solution {
public:
    vector<vector<int>> kClosest(vector<vector<int>>& points, int k) {
        // Max heap to store distances and corresponding points
        priority_queue<pair<int, vector<int>>> maxHeap;
        for (auto& point : points) {
            int distance = point[0] * point[0] + point[1] * point[1];
            maxHeap.push({distance, point});
            if (maxHeap.size() > k) maxHeap.pop(); // Remove farthest point if size > k
        }

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

OUTPUT:



Accepted 87 / 87 testcases passed

Badal1802 submitted at Mar 08, 2025 22:08

Runtime: 95 ms | Beats 31.52%

Memory: 78.06 MB | Beats 36.89%

Code | C++

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

```
        }

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

Accepted Runtime: 0 ms

points =
[[1,3],[-2,2]]

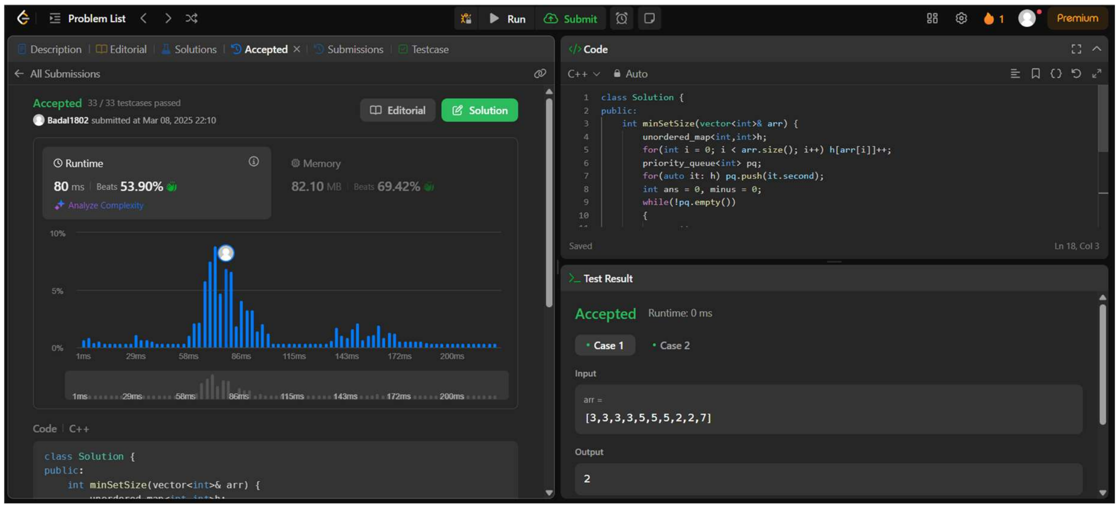
k =
1

1338.Reduce Array Size to The Half

CODE:

```cpp
class Solution {
public:
    int minSetSize(vector<int>& arr) {
        unordered_map<int,int>h;
        for(int i = 0; i < arr.size(); i++) h[arr[i]]++;
        priority_queue<int> pq;
        for(auto it: h) pq.push(it.second);
        int ans = 0, minus = 0;
        while(!pq.empty())
        {
            ans++;
            minus += pq.top();
            pq.pop();
            if(minus >= (arr.size()/2)) break;
        }
        return ans;
    }
};
```

OUTPUT:



Accepted 33 / 33 testcases passed

Badal1802 submitted at Mar 08, 2025 22:10

Runtime 80 ms | Beats 53.90%

Memory 82.10 MB | Beats 69.42%

Code | C++

```cpp
class Solution {
public:
    int minSetSize(vector<int>& arr) {
```

```cpp
class Solution {
public:
    int minSetSize(vector<int>& arr) {
        unordered_map<int,int>h;
        for(int i = 0; i < arr.size(); i++) h[arr[i]]++;
        priority_queue<int> pq;
        for(auto it: h) pq.push(it.second);
        int ans = 0, minus = 0;
        while(!pq.empty())
        {
```

Saved   Ln 18, Col 3

Test Result

Accepted Runtime: 0 ms

Case 1   Case 2

Input

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

Output

```
2
```