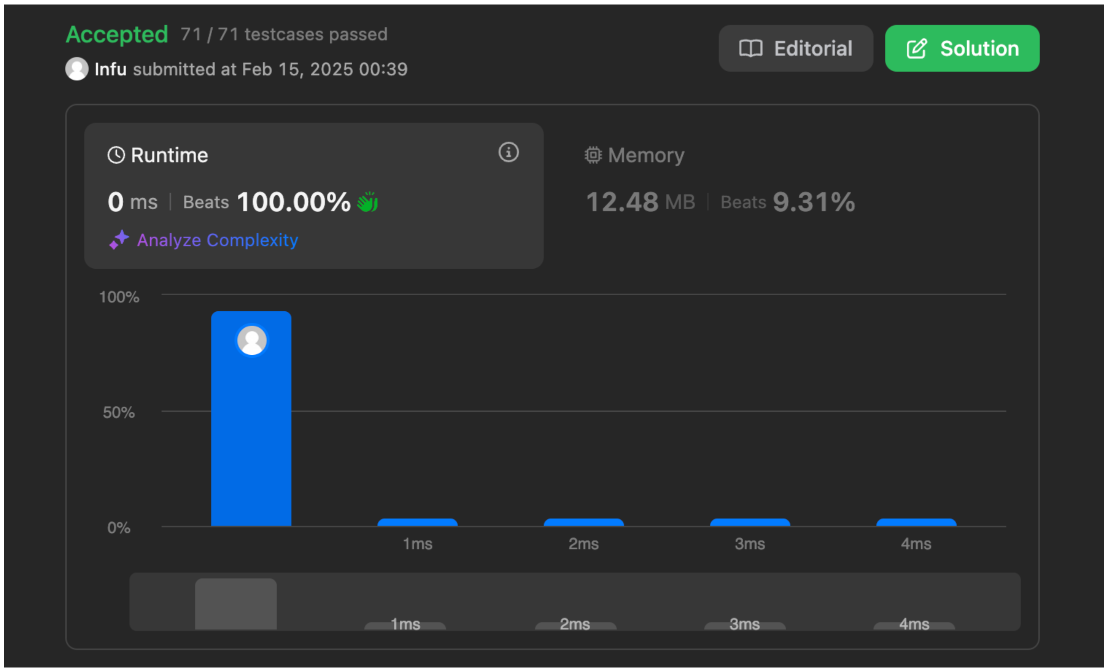
389.Find the diffrence

```
class Solution {
public:
    int findDifference(vector<int>& nums1, vector<int>& nums2) {
        unordered_set<int> set1(nums1.begin(), nums1.end()),
set2(nums2.begin(), nums2.end());
        vector<int> res1, res2;
        for (int num : set1) {
            if (set2.find(num) == set2.end()) res1.push_back(num);
        }
        for (int num : set2) {
            if (set1.find(num) == set1.end()) res2.push_back(num);
        }
        return {res1, res2};
    }
};
```



Accepted 71 / 71 testcases passed

Infu submitted at Feb 15, 2025 00:39

Runtime

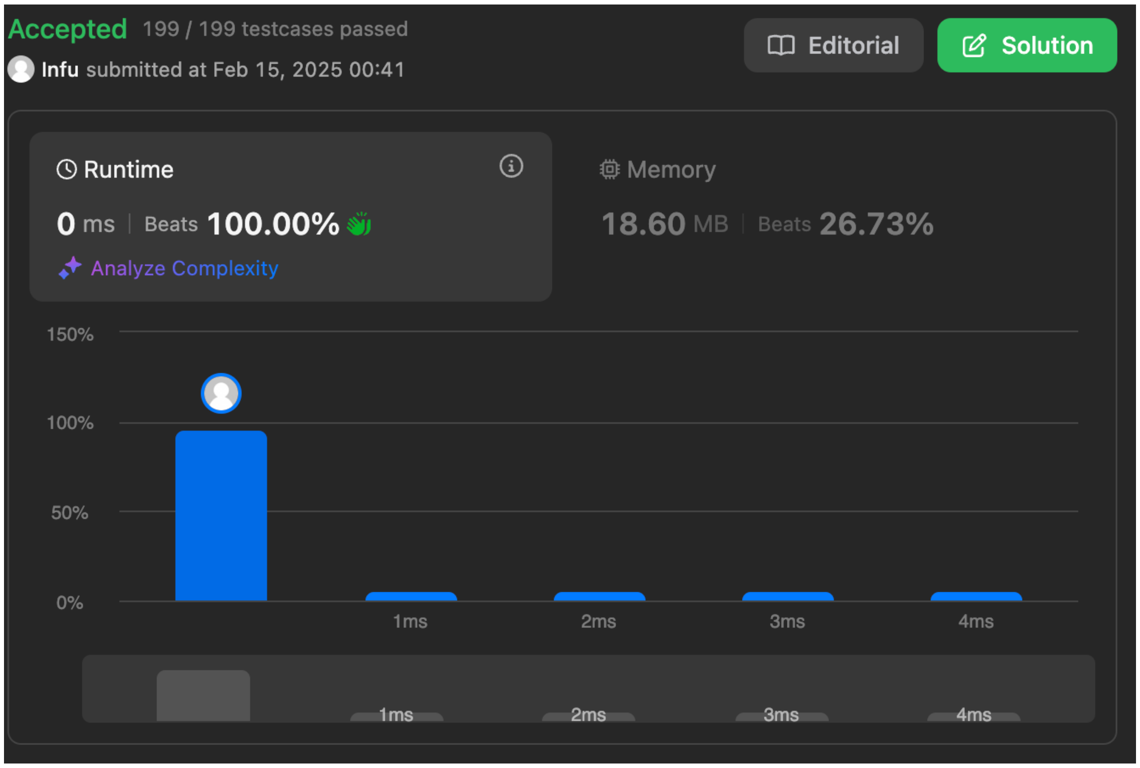
0 ms | Beats 100.00%

Memory

12.48 MB | Beats 9.31%

976.Largest Perimeter Triangle

```
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;
    }
};
```


Accepted 199 / 199 testcases passed

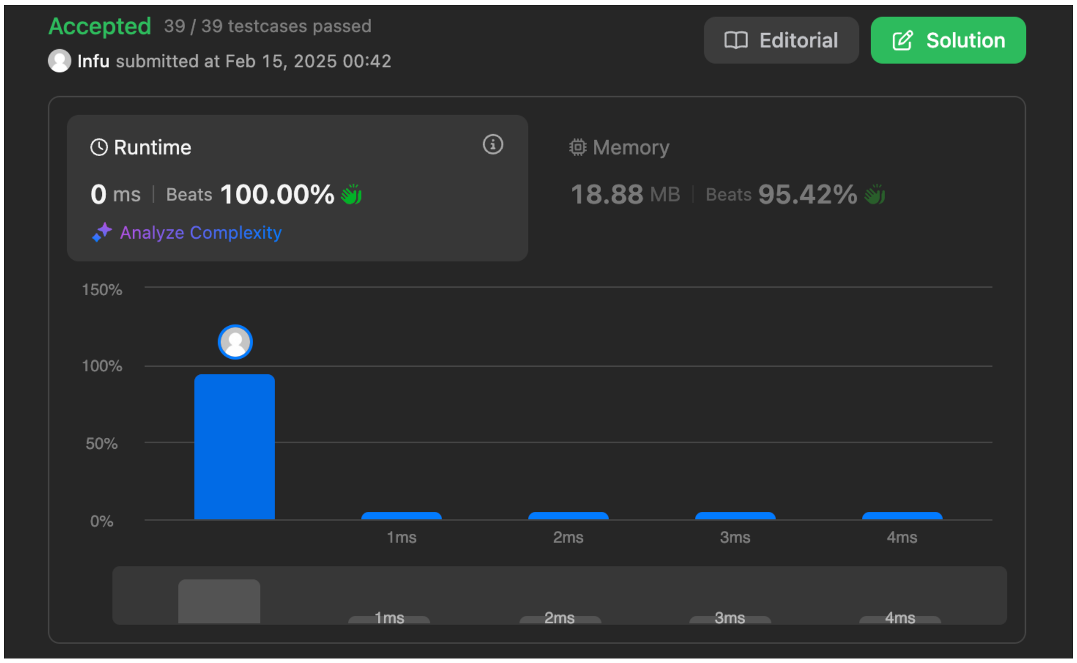
Infu submitted at Feb 15, 2025 00:41

Runtime 0 ms | Beats 100.00%

Memory 18.60 MB | Beats 26.73%

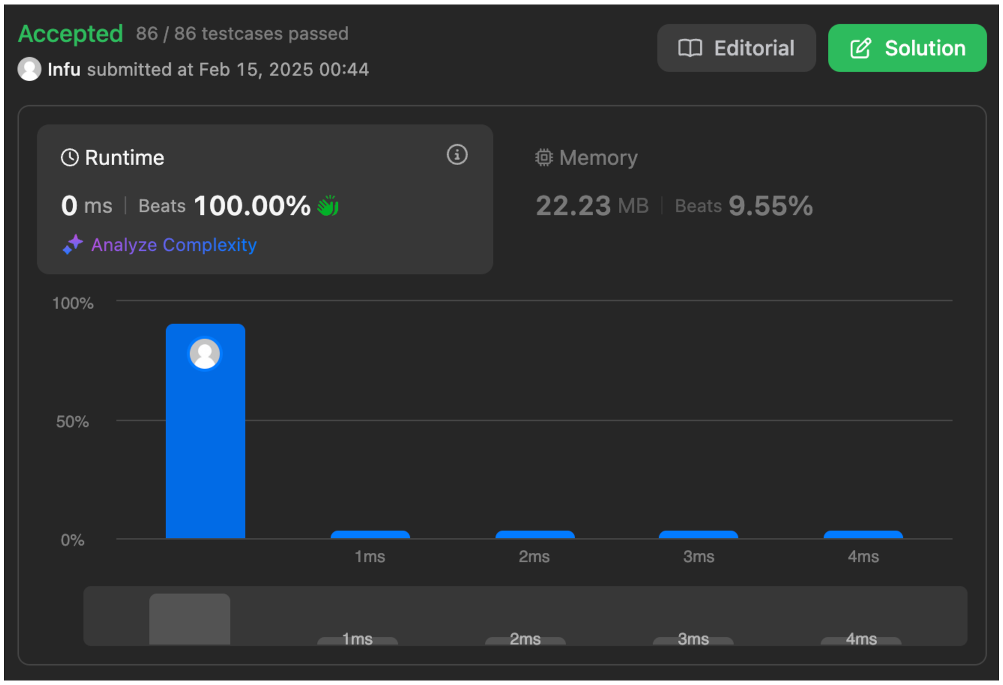
414.Third Maximum Number

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


Accepted 39 / 39 testcases passed
Infu submitted at Feb 15, 2025 00:42

451.Sort Characters By Frequency

```
class Solution {
public:
    string frequencySort(string s) {
        unordered_map<char, int> count;
        for (char c : s) count[c]++;
        priority_queue<pair<int, char>> pq;
        for (auto& p : count) pq.push({p.second, p.first});
        string res;
        while (!pq.empty()) {
            auto [freq, ch] = pq.top(); pq.pop();
            res.append(freq, ch);
        }
        return res;
    }
};
```


Accepted 86 / 86 testcases passed

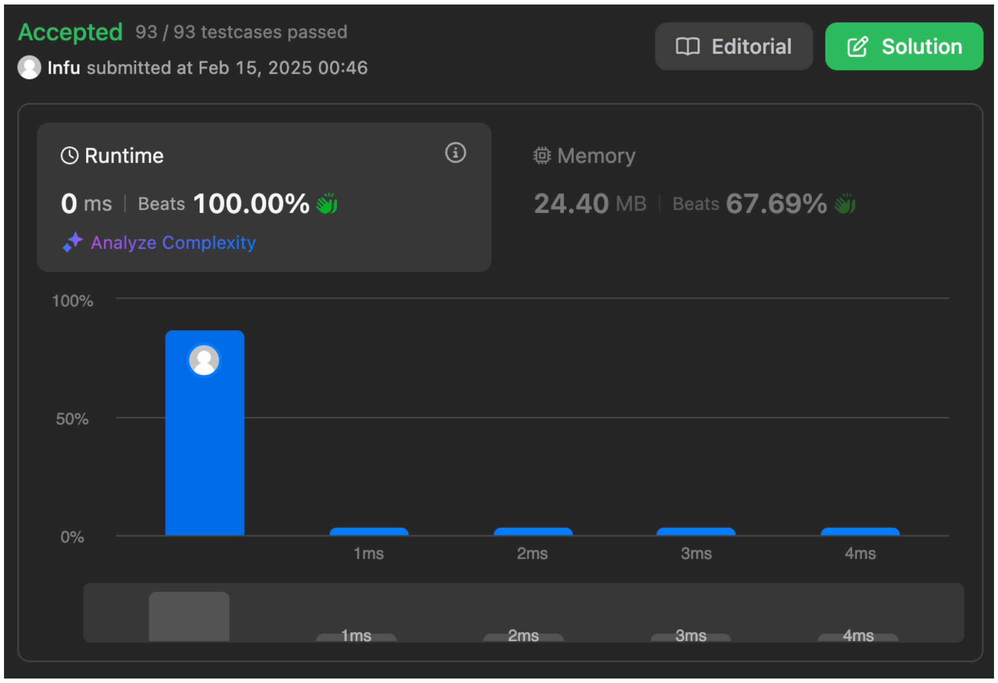
Infu submitted at Feb 15, 2025 00:44

Runtime
0 ms | Beats 100.00%

Memory
22.23 MB | Beats 9.55%

452.Minimum Number of Arrows to Burst Balloons

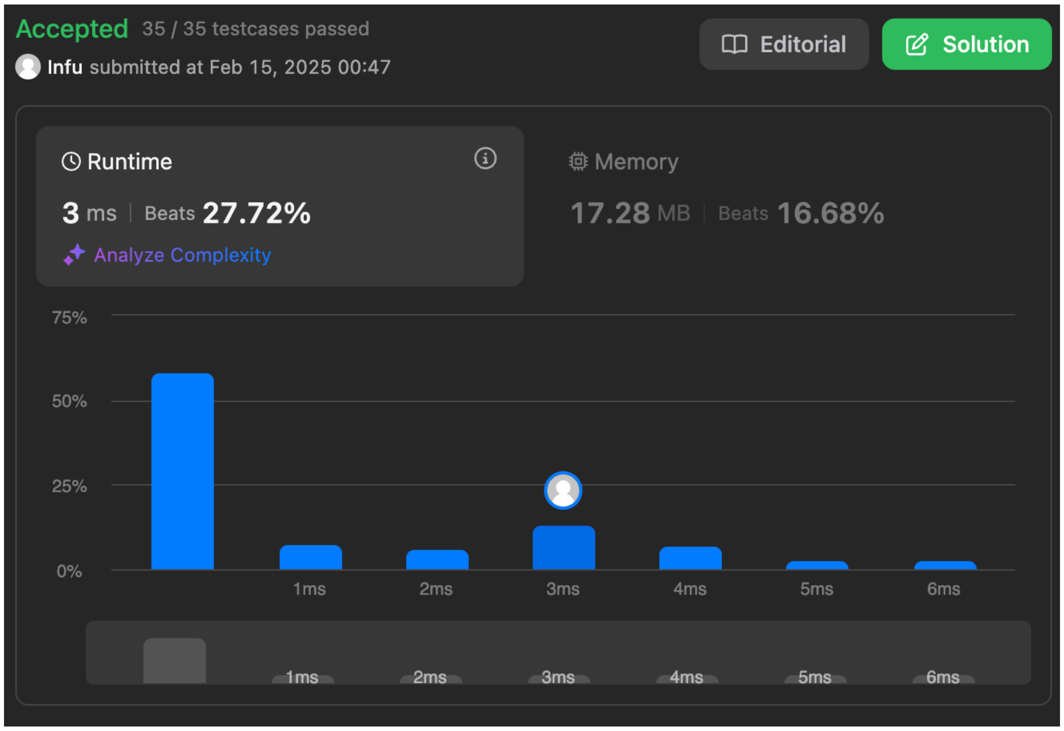
```
class Solution {
public:
    int findMinArrowShots(vector<vector<int>>& points) {
        if (points.empty()) return 0;
        sort(points.begin(), points.end(), [](auto& a, auto& b) {
return a[1] < b[1]; });
        int arrows = 1, end = points[0][1];
        for (const auto& p : points) {
            if (p[0] > end) {
                arrows++;
                end = p[1];
            }
        }
        return arrows;
    }
};
```


Accepted 93 / 93 testcases passed

Infu submitted at Feb 15, 2025 00:46

881.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, boats = 0;
        while (left <= right) {
            if (people[left] + people[right] <= limit) left++;
            right--;
            boats++;
        }
        return boats;
    }
};
```

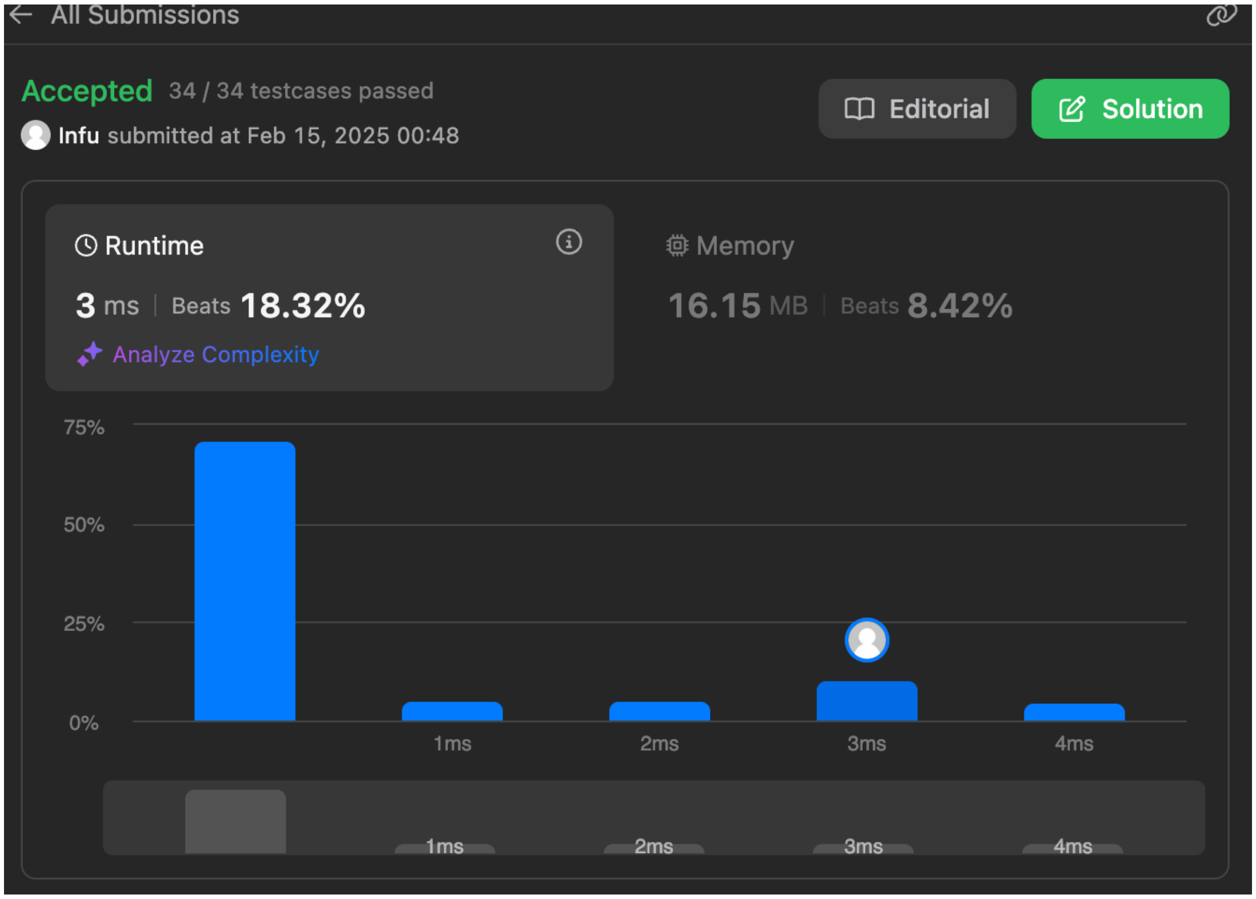

Accepted 35 / 35 testcases passed
Infu submitted at Feb 15, 2025 00:47

Runtime
3 ms | Beats 27.72%

Memory
17.28 MB | Beats 16.68%

973.K Closest Points to Origin

```
class Solution {
public:
    vector<vector<int>> kClosest(vector<vector<int>>& points, int K) {
        priority_queue<pair<double, vector<int>>> maxHeap;
        for (const auto& point : points) {
            double dist = sqrt(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;
    }
};
```



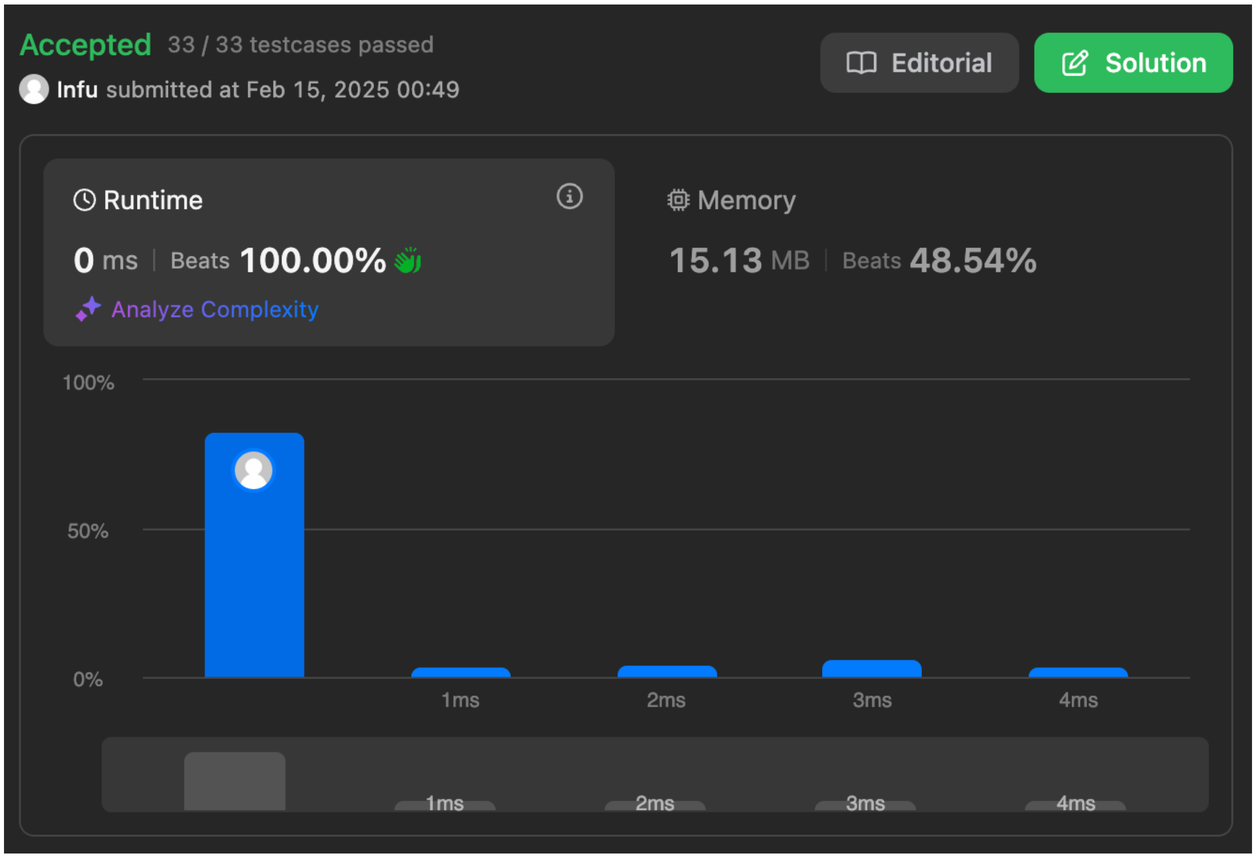
Accepted 34 / 34 testcases passed
Infu submitted at Feb 15, 2025 00:48

3 ms | Beats 18.32%

16.15 MB | Beats 8.42%

1338.Reduce Array Size to The Half

```
class Solution {
public:
    int reduceArraySize(vector<int>& arr) {
        unordered_map<int, int> count;
        for (int num : arr) count[num]++;
        priority_queue<int> pq;
        for (auto& p : count) pq.push(p.second);
        int size = 0, removed = 0;
        while (removed < arr.size() / 2) {
            removed += pq.top();
            pq.pop();
            size++;
        }
        return size;
    }
};
```



Accepted 33 / 33 testcases passed

Infu submitted at Feb 15, 2025 00:49

Runtime

0 ms | Beats 100.00%

Memory

15.13 MB | Beats 48.54%