

1) Find the Difference

Accepted × | Editorial | Solutions | <

All Submissions

Rakshita Thakur submitted at Mar 09, 2025 22:11

Runtime

358 ms | Beats 9.76%

Analyze Complexity

Memory

122.67 MB | Beats 8.22%

60%
40%
20%
0%

2ms 117ms 232ms 348ms

Code

C++ Auto

```
1 class Solution {
2 public:
3     bool isNice(const string& s) {
4         unordered_set<char> charSet(s.begin(), s.end());
5
6         for (char c : s) {
7             if (charSet.count(tolower(c)) == 0 || charSet.count(toupper(c)) == 0) {
8                 return false;
9             }
10        }
11        return true;
12    }
13
14    string longestNiceSubstring(string s) {
15        int n = s.length();
16        string longest = "";
17        for (int i = 0; i < n; i++) {
18            for (int j = i; j < n; j++) {
19                string sub = s.substr(i, j - i + 1);
20                if (isNice(sub) && sub.length() > longest.length()) {
21                    longest = sub;
22                }
23            }
24        }
25        return longest;
26    }
27 }
```

Saved

Test Result | Testcase

2) Largest Perimeter

Problem List < > x

Run Submit

Description Accepted × | Editorial | Solutions | Submissions

All Submissions

Accepted 84 / 84 testcases passed

Rakshita T... submitted at Mar 09, 2025 22:31

Runtime

20 ms | Beats 6.24%

Analyze Complexity

Memory

25.65 MB | Beats 48.92%

20%
15%
10%
5%

Code

C++ Auto

```
1 class Solution {
2 public:
3     int largestPerimeter(vector<int>& nums) {
4         int n=nums.size(),maxi=0,ans=0;
5         sort(nums.begin(),nums.end());
6         for(int i=n-1;i>=2;i--){
7             if(nums[i]<nums[i-1]+nums[i-2]){
8                 ans=nums[i]+nums[i-1]+nums[i-2];
9             }
10            maxi=max(maxi,ans);
11        }
12        return maxi;
13    }
14 }
```

Saved

Test Result | Testcase

3) Third Maximum number

The screenshot shows a code editor interface with a submission status of 'Accepted'. The left sidebar displays performance metrics: Runtime is 8 ms (Beats 5.80%) and Memory is 12.14 MB (Beats 100.00%). A bar chart below these metrics shows the user's performance relative to others. The main editor contains the following C++ code:

```
1 class Solution {
2 public:
3     int thirdMax(vector<int>& nums) {
4         int n=nums.size();
5         vector<int> ans;
6
7         sort(nums.begin(),nums.end());
8         for(int i=0;i+1<n;i++){
9             if(nums[i]!=nums[i+1]){
10                 ans.push_back(nums[i]);
11             }
12         }
13         ans.push_back(nums[n-1]);
14         int n2=ans.size();
15         int max=ans[n2-1];
16         if(n2==2||n2==1){
17             return max;
18         }
19         else{
20             max= ans[n2-3];
21         }
22     }
23 }
```

The bottom of the interface shows tabs for 'Description', 'Editorial', 'Solutions', and 'Submissions', along with 'Test Result' and 'Testcase' buttons.

4) Sort Character by Frequency

The screenshot shows a code editor interface with a submission status of 'Accepted'. The left sidebar displays performance metrics: Runtime is 5 ms (Beats 28.82%) and Memory is 10.71 MB (Beats 87.68%). A bar chart below these metrics shows the user's performance relative to others. The main editor contains the following C++ code:

```
1 class Solution {
2 public:
3     string frequencySort(string s) {
4         map<char,int> ans;string r;
5         for(int i=0;i<s.length();i++){
6             ans[s[i]]++;
7         }
8         vector<pair<int,char>> vec;
9         for(auto x:ans){
10             vec.push_back({x.second,x.first});
11         }
12         sort(vec.rbegin(),vec.rend());
13         for(auto x:vec){
14             int ix=x.first;
15             while(ix){
16                 r+=x.second;
17                 ix--;
18             }
19         }
20         return r;
21     }
22 }
```

The bottom of the interface shows tabs for 'Description', 'Editorial', 'Solutions', and 'Submissions', along with 'Test Result' and 'Testcase' buttons.

5) Boat to Save People

The screenshot shows the LeetCode interface for the 'Boat to Save People' problem. The solution is in C++ and uses a two-pointer approach. The runtime is 14 ms, beating 77.88% of other solutions. The memory usage is 45.78 MB, beating 68.75% of other solutions. A bar chart at the bottom shows the distribution of runtime times, with a peak around 10-15 ms.

Problem List < >

Description | Accepted x | Editorial | Solutions | Submissions

All Submissions

Accepted 78 / 78 testcases passed Solution

Rakshita T... submitted at Mar 09, 2025 22:38

Runtime

14 ms | Beats 77.88%

 Analyze Complexity

Memory

45.78 MB | Beats 68.75%

15%
10%
5%

Code

```
1 class Solution {
2 public:
3     int numRescueBoats(vector<int>& people, int limit) {
4         sort(people.begin(), people.end());
5         int i = 0, j = people.size() - 1;
6         int boats = 0;
7
8         while (i <= j) {
9
10            if (people[i] + people[j] <= limit) {
11                i++;
12            }
13
14            j--;
15            boats++;
16        }
17
18        return boats;
19    }
20 };
```

Saved

 Test Result | Testcase

6) Reduce Array Size by half

The screenshot shows the LeetCode interface for the 'Reduce Array Size by half' problem. The solution is in C++ and uses a frequency map and sorting to reduce the array size by half. The runtime is 95 ms, beating 33.81% of other solutions. The memory usage is 85.43 MB, beating 25.71% of other solutions. A bar chart at the bottom shows the distribution of runtime times, with a peak around 70-80 ms.

Problem List < >

Description | Editorial | Accepted x | Solutions | Submissions

All Submissions

Rakshita T... submitted at Mar 09, 2025 22:41

Runtime

95 ms | Beats 33.81%

 Analyze Complexity

Memory

85.43 MB | Beats 25.71%

10%
5%
0%

Code

```
1 class Solution {
2 public:
3     int minSetSize(vector<int>& arr) {
4         unordered_map<int, int> freq;
5         for (int num : arr) {
6             freq[num]++;
7         }
8
9         vector<int> counts;
10        for (auto& pair : freq) {
11            counts.push_back(pair.second);
12        }
13
14        sort(counts.rbegin(), counts.rend());
15
16        int removed = 0, size = arr.size(), target = size / 2;
17        for (int count : counts) {
18            removed += count;
19        }
20    }
```

Saved

 Test Result | Testcase

</> Source