ASSIGNMENT 5

Student Name: Mehak Chauhan **UID: 22BCS12675**
Branch: CSE **Section: 22BCS_IOT_605 B**
Semester: 6th **DOP:05-03-2025**
Subject: Advanced Programming Lab-II **Subject Code: 22CSP-351**

Question 1

389. Find the Difference

Easy | Topics | Companies

You are given two strings `s` and `t`.

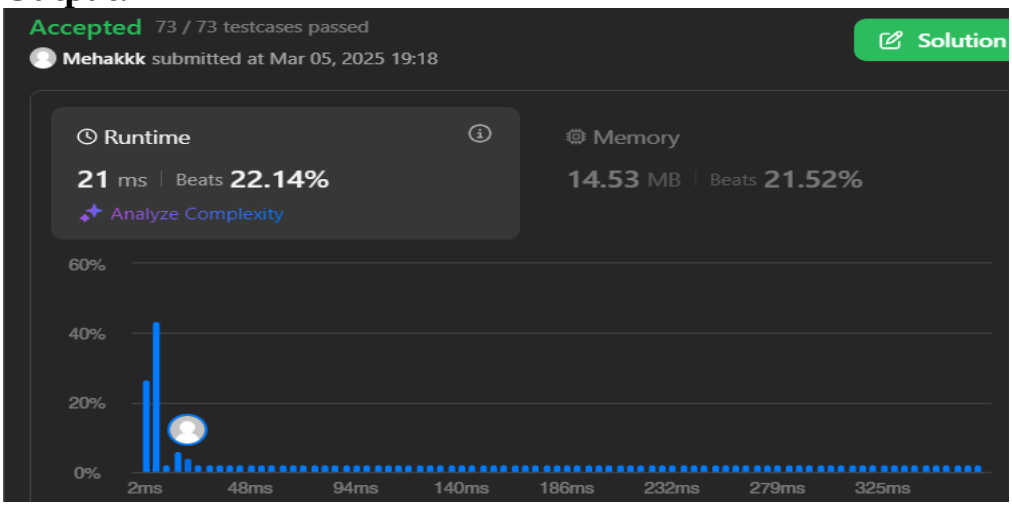
String `t` is generated by random shuffling string `s` and then add one more letter at a random position.

Return the letter that was added to `t`.

Code:

```cpp
class Solution {
public:
    char findTheDifference(string s, string t) {
        int s_sum = 0, t_sum = 0;
        for (char c : s) {
            s_sum += int(c);
        }
        for (char c : t) {
            t_sum += int(c);
        }
        return char(t_sum - s_sum);
    }
};
```

Output:



Question 2

976. Largest Perimeter Triangle

Solved

Easy | Topics | Companies

Given an integer array `nums`, return *the largest perimeter of a triangle with a non-zero area, formed from three of these lengths*. If it is impossible to form any triangle of a non-zero area, return `0`.

Code:

```cpp
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

Mehakkk submitted at Mar 05, 2025 19:53

Editorial | Solution

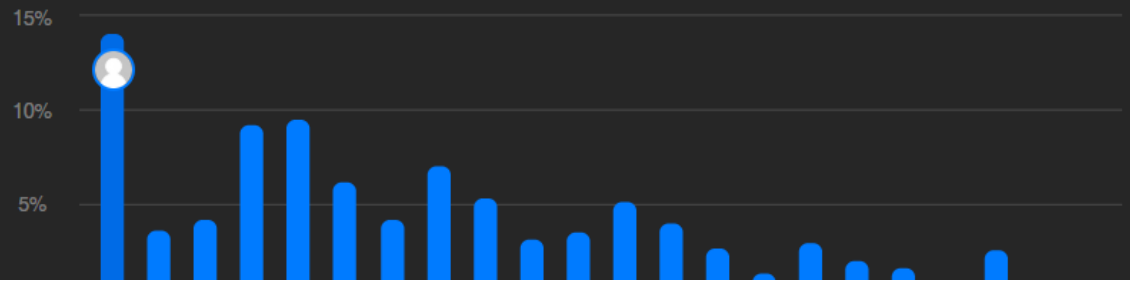
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

25.61 MB | Beats 48.68%



Question 3

414. Third Maximum Number

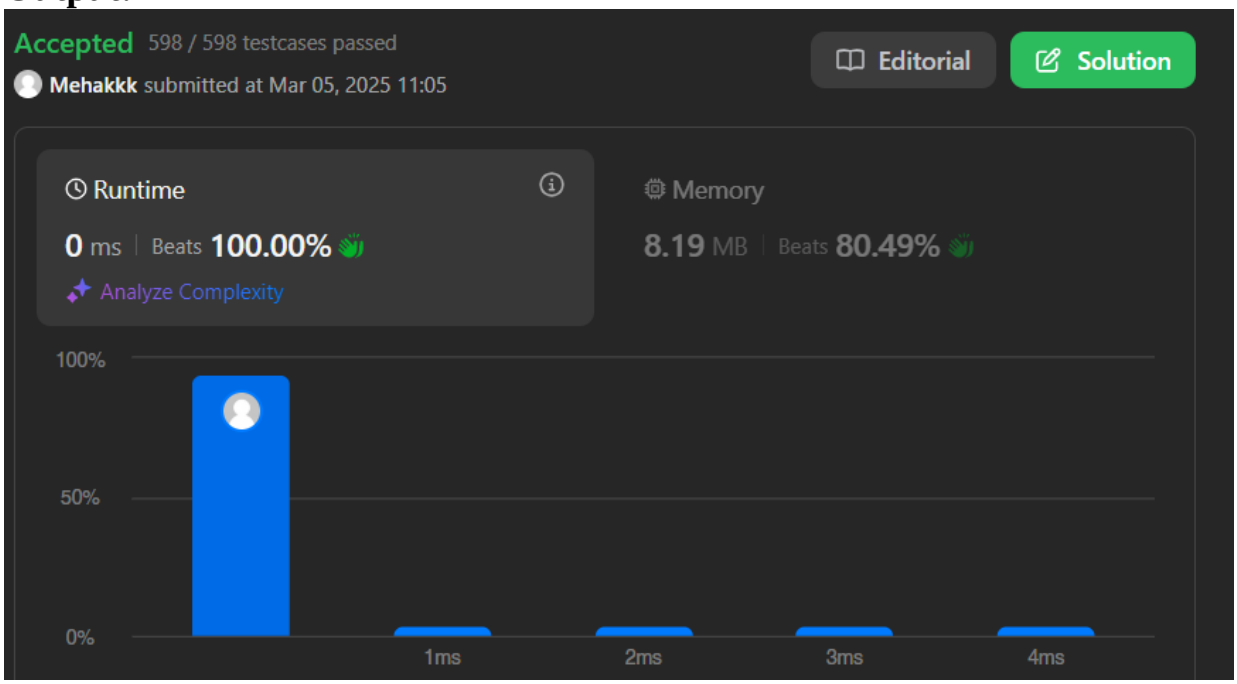
Easy | Topics | Companies

Given an integer array `nums`, return *the **third distinct maximum** number in this array. If the third maximum does not exist, return the **maximum** number.*

Code:

```
class Solution {
public:
    int thirdMax(vector<int>& nums) {
        sort(nums.begin(),nums.end());
        int largest,seclargest,thirdlargest;
        largest= nums[0];
        seclargest=nums[0];
        thirdlargest=nums[0];
        for(int i=0;i<nums.size();i++){
            if(nums[i]>largest){
                thirdlargest=seclargest;
                seclargest=largest;
                largest=nums[i];
            }
            else if(nums[i]>seclargest && nums[i]<largest){
                thirdlargest=seclargest;
                seclargest=nums[i];
            }
            else if(nums[i]>thirdlargest && nums[i]<seclargest){
                thirdlargest=nums[i];
            }
        }
        return ((nums.size()<=2 || seclargest==thirdlargest)?largest:thirdlargest);
    }
};
```

Output:



Question 4

451. Sort Characters By Frequency

Solved

Medium | Topics | Companies

Given a string `s`, sort it in **decreasing order** based on the **frequency** of the characters. The **frequency** of a character is the number of times it appears in the string.

Return *the sorted string*. If there are multiple answers, return *any of them*.

Code:

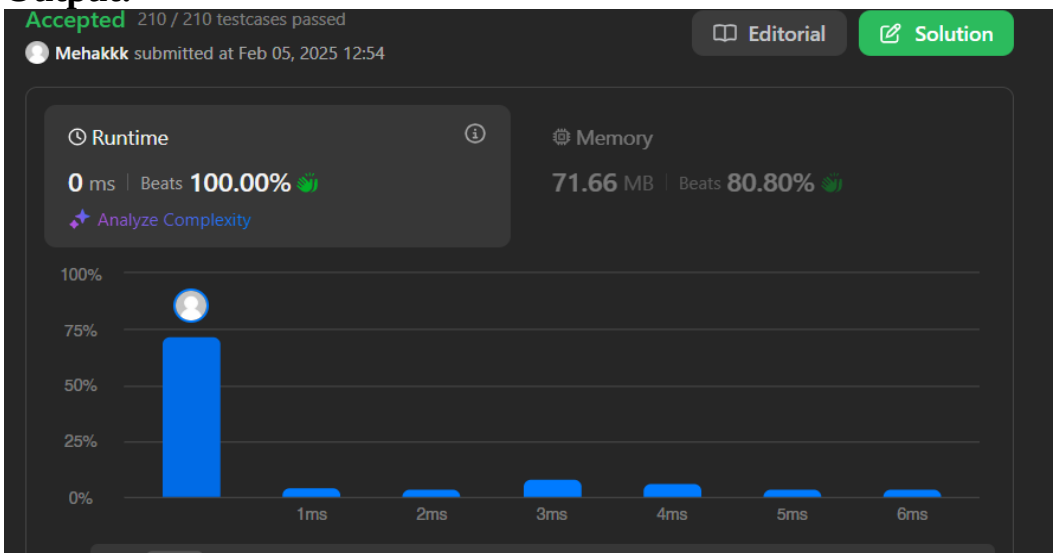
```
class Solution {
public:
    string frequencySort(string s) {
        unordered_map<char,int> map;
        vector<pair<int,char>>v;
        string res;
        for(int i=0;i<s.length();i++){
            map[s[i]]++;
        }

        for(auto it: map){
            v.push_back({it.second,it.first});
        }
        sort(v.rbegin(),v.rend());

        for(auto it : v){
            for(int j=0;j<it.first;j++){
                res+=it.second;
            }
        }
```

```
        return res;
    }
};
```

Output:



Question 5

881. Boats to Save People

Solved

Medium | Topics | Companies

You are given an array `people` where `people[i]` is the weight of the i^{th} person, and an **infinite number of boats** where each boat can carry a maximum weight of `limit`. Each boat carries at most two people at the same time, provided the sum of the weight of those people is at most `limit`.

Return *the minimum number of boats to carry every given person.*

Code:

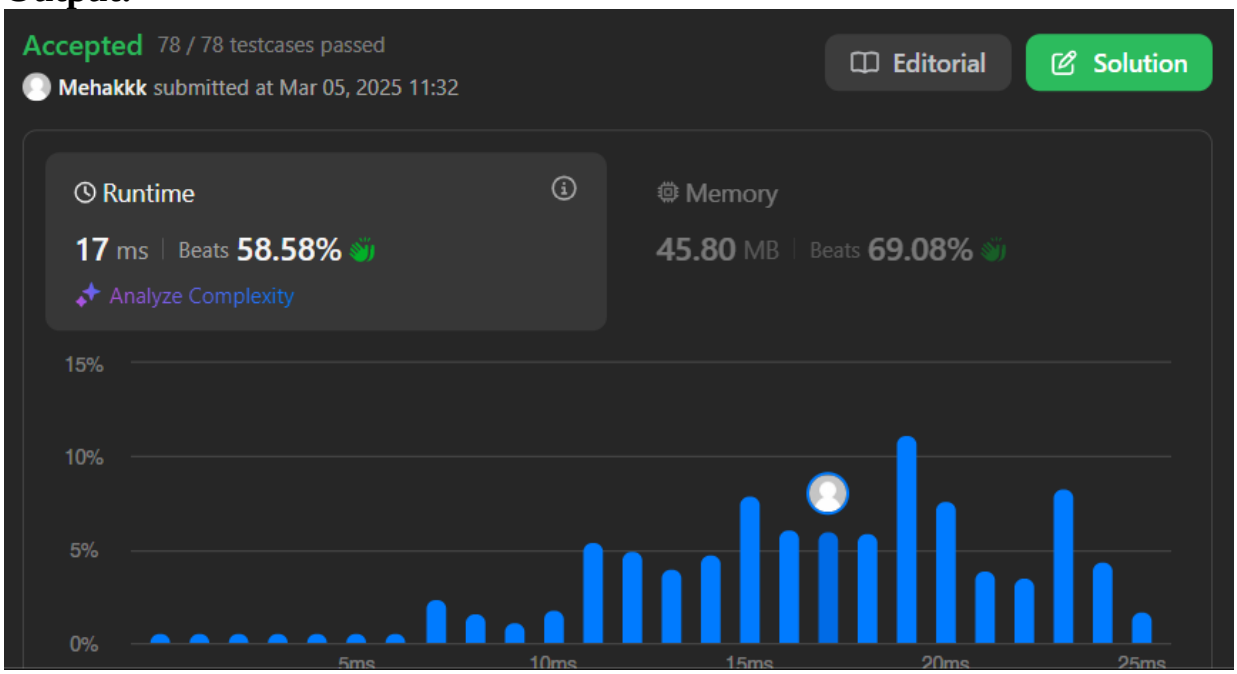
```
class Solution {
public:
    int numRescueBoats(vector<int>& people, int limit) {
        sort(people.begin(), people.end());

        int i=0, boats=0;
        int n= people.size()-1;

        while(i<=n){
            if(people[i]+people[n]<=limit){
                i++;
                n--;
                boats++;
            }
            else{
                boats++;
                n--;
```

```
            }
        }
        return boats;
    }
};
```

Output:



Question 6

1338. Reduce Array Size to The Half

Medium | Topics | Companies | Hint

You are given an integer array `arr`. You can choose a set of integers and remove all the occurrences of these integers in the array.

Return *the minimum size of the set so that* ***at least*** *half of the integers of the array are removed.*

Code

```
class Solution {
public:
    int minSetSize(vector<int>& arr)
    {
        map<int,int>mp;
        for(auto val:arr) mp[val]++;

        priority_queue<int>pq;
        for(auto [val, cnt]:mp)
            pq.push(cnt);

        int ans = 0, need = arr.size()/2;
        while(need > 0)
```

```
        {
            ans++;
            need -= pq.top(); pq.pop();
        }
        return ans;
    }
};
```

Output