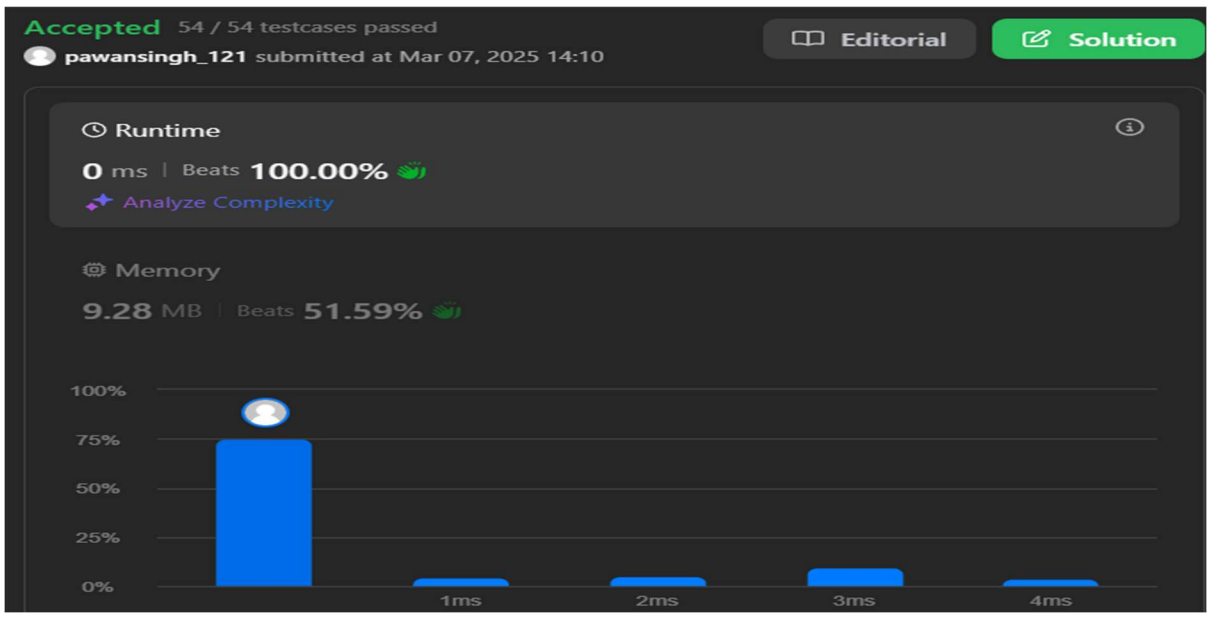
ASSIGNMENT -5 (ADVANCED PROGRAMMING)
NAME:PAWAN SINGH
UID:22BCS14871
SEC:FL_IOT_602/A

389.Find the diffrence

```
class Solution {
public:
    char findTheDifference(string s, string t)
    {
      for(int i=0;i<s.size();i++)
        t[i+1]+=t[i]-s[i];
      return t[t.size()-1];
    }
};
```

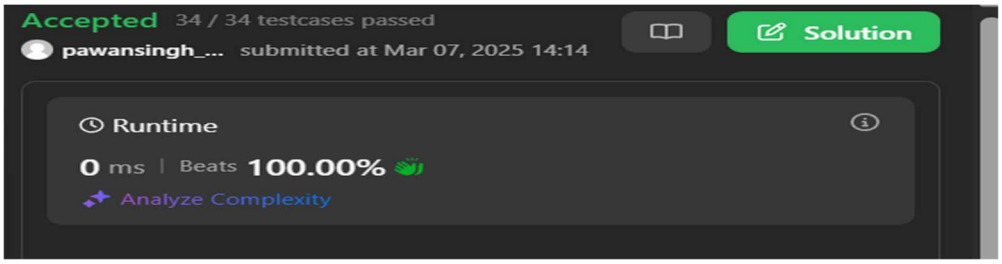


976.Largest Perimeter Triangle

```
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;
    }
};
```

414.Third Maximum Number

```
class Solution {
public:
    int thirdMax(vector<int>& nums) {
        sort(nums.begin(),nums.end());
        int count=0;
        int third_maximum=0;
        for(int i=nums.size()-1 ;i>0;i--){
            if(nums[i]!=nums[i-1]){
                count++;
               third_maximum=nums[i];  }
           else if(i==1 && nums[i]==nums[i-1]){
                count++;      third_maximum=nums[i];}
            if(count>2){
                return third_maximum;
           }}
        if(count+1==3 && nums[0]!=nums[1]){
           return nums[0]; }
        return nums[nums.size()-1];  }};
```

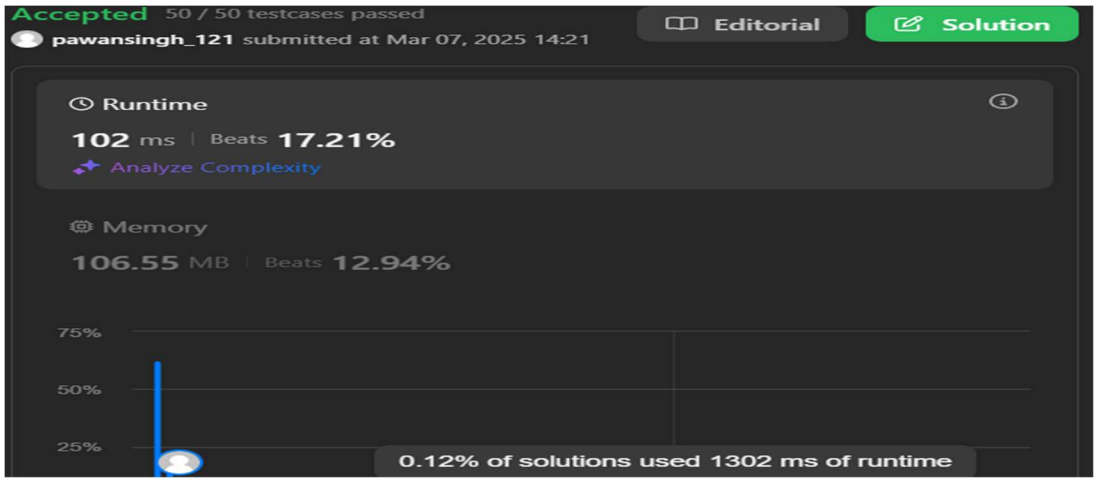


451.Sort Characters By Frequency

```
class Solution {
public:
    static bool st(pair<char,int>& a,pair<char,int>& b)
    {
        if (a.second == b.second)  return a.first < b.first;
        return a.second > b.second;}
    string frequencySort(string s) {
        unordered_map<char,int> mp;
        for(char c:s)
        {mp[c]++;}
        vector<pair<char,int>> arr(mp.begin(),mp.end());
        sort(arr.begin(),arr.end(),st);
        string s1;
        for(auto& it:arr)
        {for(int i=0;i<it.second;i++)
            {  s1+=it.first; }}
        return s1;
    }};
```

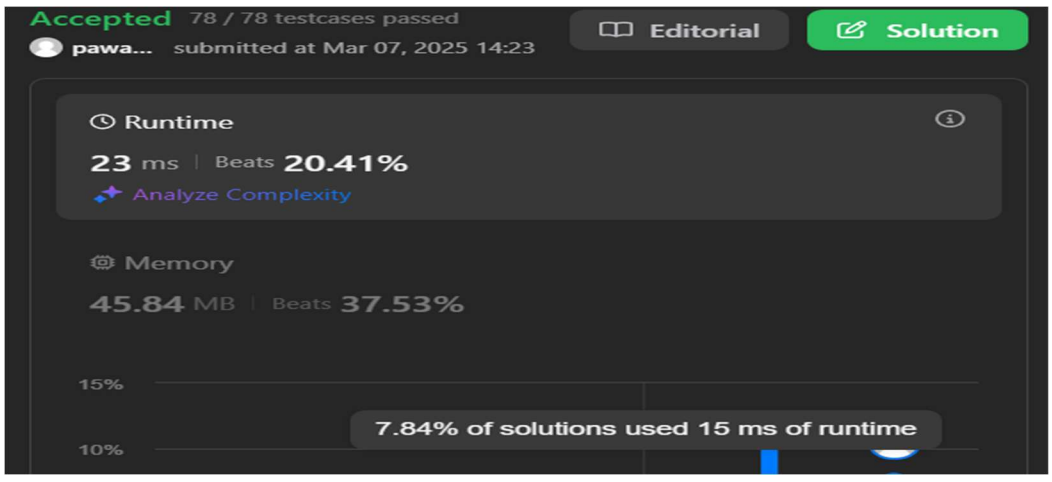
452.Minimum Number of Arrows to Burst Balloons

```
class Solution {
public:
    int findMinArrowShots(vector<vector<int>>& p) {
        sort(p.begin(), p.end());
        int lastpoint = p[0][1];
        int ans = 1;
        for(auto point : p) {
            if(point[0] > lastpoint) {
                ans++;
                lastpoint = point[1];
            }
            lastpoint = min(point[1],lastpoint);
        }
        return ans;
    }
};
```



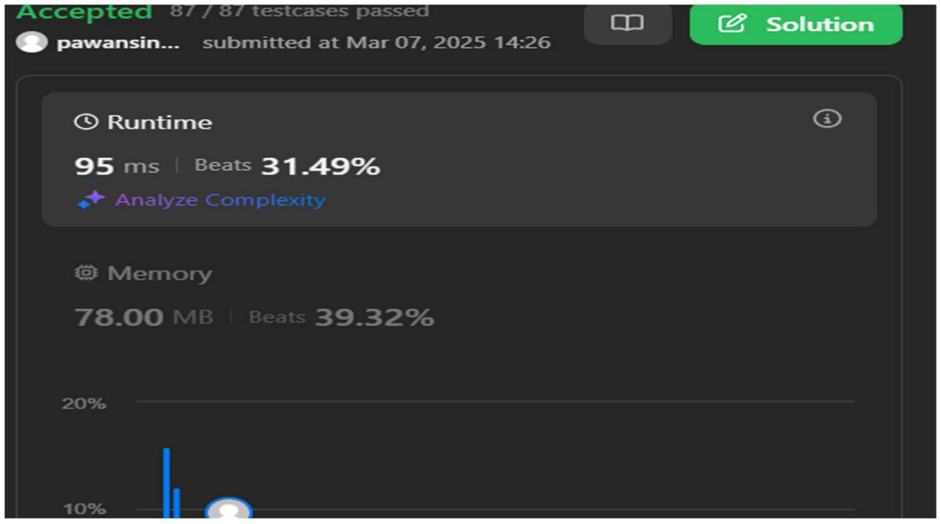
881.Boats to Save People

```
class Solution {
public:
    int numRescueBoats(vector<int>& people, int limit) {
        sort(people.begin() , people.end());
        int i=0 , boats=0;
        int j=people.size()-1;
        while(i<=j){
            if(people[i]+people[j]<=limit){
                i++;
                j--;
                boats++;
            }
            else{
                boats++;
                j--;
            }
        } return boats;
    }};
```



973.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& point : points) {
            int distance = point[0] * point[0] + point[1] * point[1];
            maxHeap.push({distance, point});
            if (maxHeap.size() > k) maxHeap.pop();
        }
        vector<vector<int>> ans;
        while (!maxHeap.empty()) {
            ans.push_back(maxHeap.top().second);
            maxHeap.pop();
        }
        return ans;
    }
};
```



1338.Reduce Array Size to The Half

```
class Solution {
public:
    int minSetSize(vector<int>& arr) {
        unordered_map<int, int> counter;
        priority_queue<int> q;
        int res = 0, removed = 0;
        for (auto a : arr) counter[a]++;
        for (auto c : counter) q.push(c.second);
        while (removed < arr.size() / 2) {
            removed += q.top();
            q.pop();
            res++;
        }
        return res;
    }
};
```