Sanskar Saurav

20BCS2000

MM 903 DDWC 43(IT Skills)

Worksheet- 3

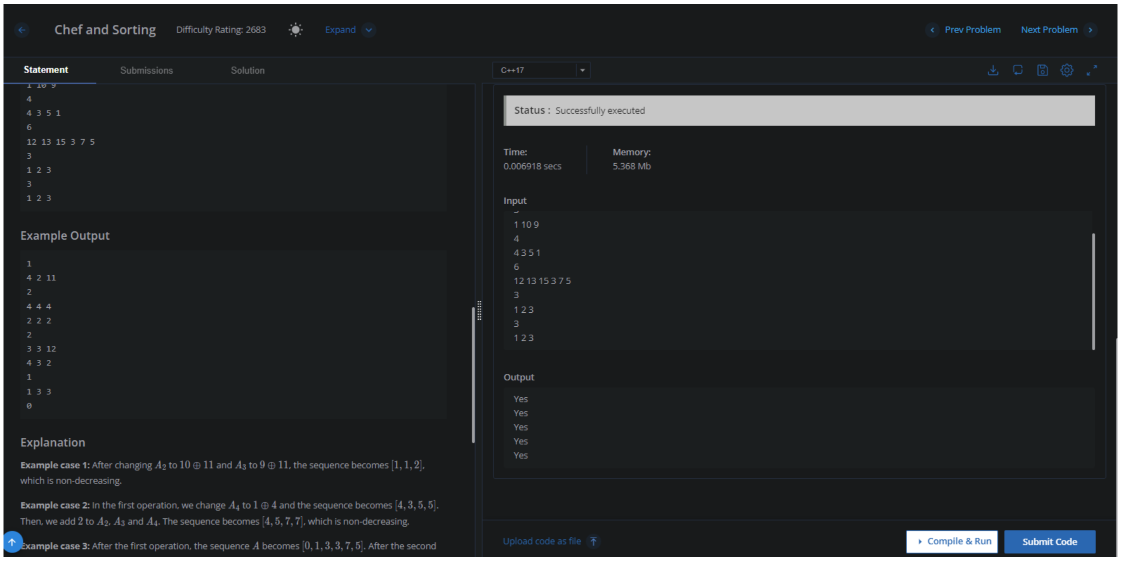
Q1. Chef and sorting

Code:

```
#include <iostream>
using namespace std;

int main() {
                    int t,cnt=0;
        cin >>t;
        while(t--){
            int n;
            cin >>n;
            int arr[n*2];
            for(int i=0;i<(n*2);i++){
                cin >>arr[i];
            }
            for(int i=0;i<(n*2)-1;i++){
                cnt=0;

                for(int j=i;j<n*2;j++){
```

```
             if(arr[i]==arr[j]){
               cnt++;
             }
            }
            if(cnt==3){
              cout <<"No\n";
              break;
            }
          }
          if(cnt<3){
            cout <<"Yes\n";
          }
        }
        return 0;
}
```

OUTPUT:



```
1 10 9
4
4 3 5 1
6
12 13 15 3 7 5
3
1 2 3
3
1 2 3
```

Example Output

```
1
4 2 11
2
4 4 4
2 2 2
2
3 3 12
4 3 2
1
1 3 3
0
```

Explanation

Example case 1: After changing A_2 to $10 \oplus 11$ and A_3 to $9 \oplus 11$, the sequence becomes $[1, 1, 2]$, which is non-decreasing.

Example case 2: In the first operation, we change A_4 to $1 \oplus 4$ and the sequence becomes $[4, 3, 5, 5]$. Then, we add 2 to A_2, A_3 and A_4. The sequence becomes $[4, 5, 7, 7]$, which is non-decreasing.

Example case 3: After the first operation, the sequence A becomes $[0, 1, 3, 3, 7, 5]$. After the second

Status : Successfully executed

Time:
0.006918 secs

Memory:
5.368 Mb

Input

```
1 10 9
4
4 3 5 1
6
12 13 15 3 7 5
3
1 2 3
3
1 2 3
```

Output

```
Yes
Yes
Yes
Yes
Yes
```

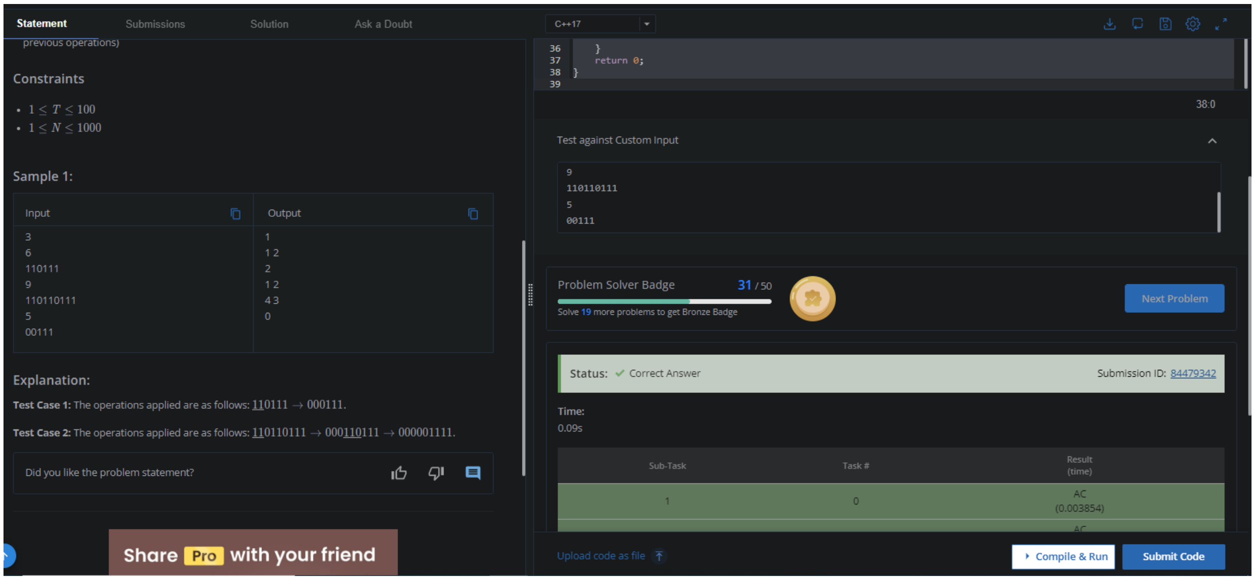
Q2. Flip Sorting

Code:

```
#include <iostream>
using namespace std;
#include<bits/stdc++.h>
int main() {
    // your code goes here
    int t;
    cin>>t;
    while(t--){
      int n;
      cin>>n;
      vector<int> vec;
      int count=0;
      string str;
      cin>>str;
      for(int i=n-1;i>=0;i--){
        if(str[i]=='0'){
          count++;
          vec.push_back(i);
           for(int j=0;j<=i;j++){
          if(str[j]=='1'){
            str[j]='0';
          }else{
            str[j]='1';
```

```
              }

            }

            }



          }



          cout<<count<<endl;

          for(auto x: vec){

            cout<<1<<" "<<x+1<<endl;

          }




        }

        return 0;

}
```

OUTPUT:



previous operations)

Constraints

- $1 \leq T \leq 100$
- $1 \leq N \leq 1000$

Sample 1:

Input	Output

3
6
110111
9
110110111
5
00111
	1
	1 2
	2
	1 2
	4 3
	0

Explanation:

Test Case 1: The operations applied are as follows: $\underline{11}0111 \rightarrow 000111$.

Test Case 2: The operations applied are as follows: $\underline{11}0110111 \rightarrow 000\underline{110}111 \rightarrow 000001111$.

Did you like the problem statement?

```
    }
    return 0;
}
```

38:0

Test against Custom Input

```
9
110110111
5
00111
```

Problem Solver Badge 31 / 50

Solve 19 more problems to get Bronze Badge

Status: ✓ Correct Answer   Submission ID: 84479342

Time:
0.09s

Sub-Task	Task #	Result (time)
1	0	AC (0.003854)
		AC

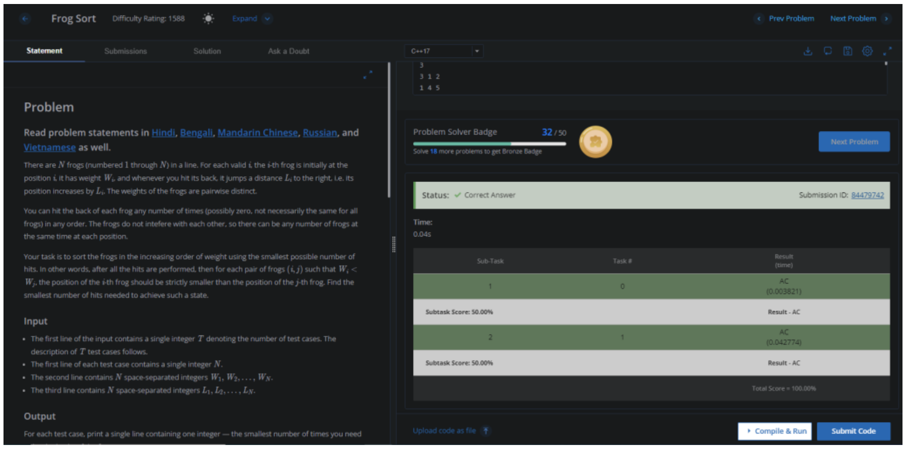
Q3. Frog sort

Code:

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main() {
        // your code goes here
        int t,n,w[4], l[4];
        cin>>t;
        while(t--){
          cin>>n;
          for(int i=0; i<n; i++){
            cin>>w[i];
          }
          for(int i=0; i<n; i++){
            cin>>l[i];
          }

          vector<pair<int, int>> frog;
          for(int i=0; i<n; i++){
            frog.push_back({w[i], i+1});
          }
          sort(frog.begin(), frog.end());
```

```
      int ans = 0;
      for(int i=1; i<n; i++){
        int pos = frog[i].second;
        int prev = frog[i-1].second;


        int steps = l[pos-1];


        while(pos <= prev){
          pos += steps;
          ans++;
        }
        frog[i].second = pos;


      }
      cout << ans << endl;
    }
    return 0;
}
```

:



Problem

Read problem statements in Hindi, Bengali, Mandarin Chinese, Russian, and Vietnamese as well.

There are N frogs (numbered 1 through N) in a line. For each valid i, the i-th frog is initially at the position i, it has weight W_i, and whenever you hit its back, it jumps a distance L_i to the right, i.e. its position increases by L_i. The weights of the frogs are pairwise distinct.

You can hit the back of each frog any number of times (possibly zero, not necessarily the same for all frogs) in any order. The frogs do not intefere with each other, so there can be any number of frogs at the same time at each position.

Your task is to sort the frogs in the increasing order of weight using the smallest possible number of hits. In other words, after all the hits are performed, then for each pair of frogs (i, j) such that $W_i < W_j$, the position of the i-th frog should be strictly smaller than the position of the j-th frog. Find the smallest number of hits needed to achieve such a state.

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first line of each test case contains a single integer N.
- The second line contains N space-separated integers $W_1, W_2, \ldots, W_N$.
- The third line contains N space-separated integers $L_1, L_2, \ldots, L_N$.

Output

For each test case, print a single line containing one integer — the smallest number of times you need

```
3
3 1 2
1 4 5
```

Status: ✓ Correct Answer Submission ID: 84479742

Time:
0.04s

Sub-Task	Task #	Result (time)
1	0	AC (0.003821)
Subtask Score: 50.00%		**Result - AC**
2	1	AC (0.042774)
Subtask Score: 50.00%		**Result - AC**
		Total Score = 100.00%

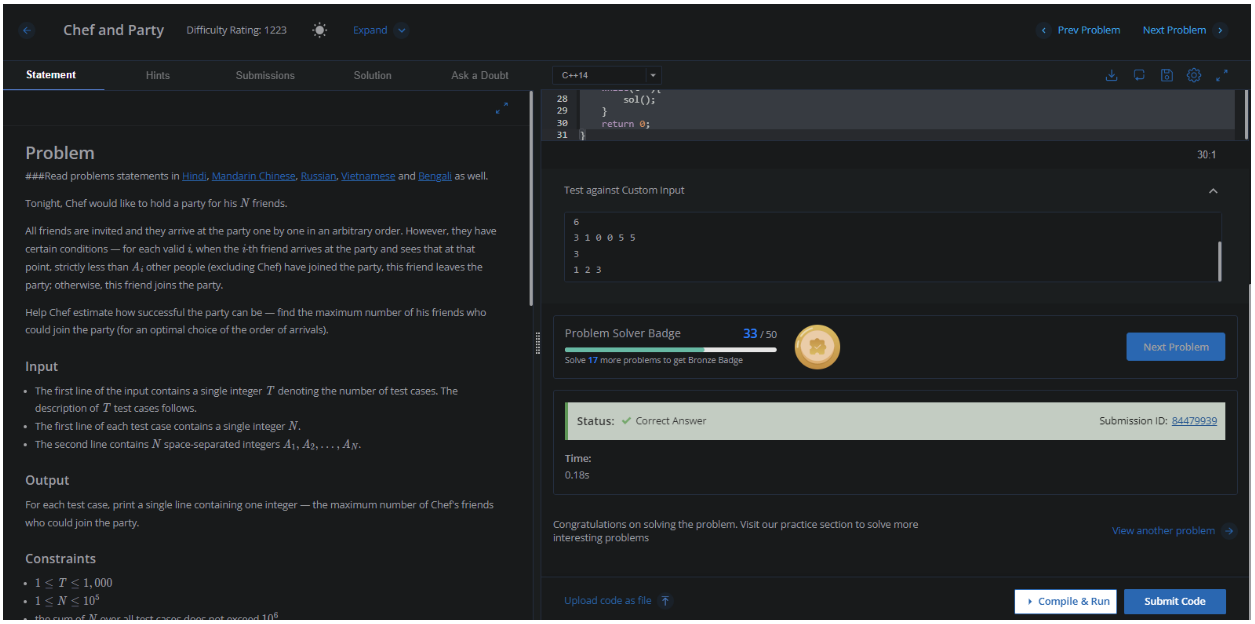
Q4. Chef and Party

Code:

```
#include <bits/stdc++.h>
using namespace std;


 void sol(){
     int n;
     cin>>n;
     int a[n];
     for(int i=0;i<n;i++){
         cin>>a[i];
     }
     int count=0;
     sort(a,a+n);
     for(int i=0;i<n;i++){
         if(a[i]<=count){
             count++;
         }
     }
     cout<<count<<endl;
 }
```

```
int main() {
        int t;
        cin>>t;
        while(t--){
          sol();
        }
        return 0;
}
```

OUTPUT



Problem

###Read problems statements in Hindi, Mandarin Chinese, Russian, Vietnamese and Bengali as well.

Tonight, Chef would like to hold a party for his N friends.

All friends are invited and they arrive at the party one by one in an arbitrary order. However, they have certain conditions — for each valid i, when the i-th friend arrives at the party and sees that at that point, strictly less than A_i other people (excluding Chef) have joined the party, this friend leaves the party; otherwise, this friend joins the party.

Help Chef estimate how successful the party can be — find the maximum number of his friends who could join the party (for an optimal choice of the order of arrivals).

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first line of each test case contains a single integer N.
- The second line contains N space-separated integers $A_1, A_2, \ldots, A_N$.

Output

For each test case, print a single line containing one integer — the maximum number of Chef's friends who could join the party.

Constraints

- $1 \leq T \leq 1,000$
- $1 \leq N \leq 10^5$
- the sum of N over all test cases does not exceed 10^6

```
        sol();
    }
    return 0;
}
```

30:1

Test against Custom Input

```
6
3 1 0 0 5 5
3
1 2 3
```

Problem Solver Badge 33 / 50

Solve 17 more problems to get Bronze Badge

Status: ✓ Correct Answer   Submission ID: 84479939

Time:
0.18s

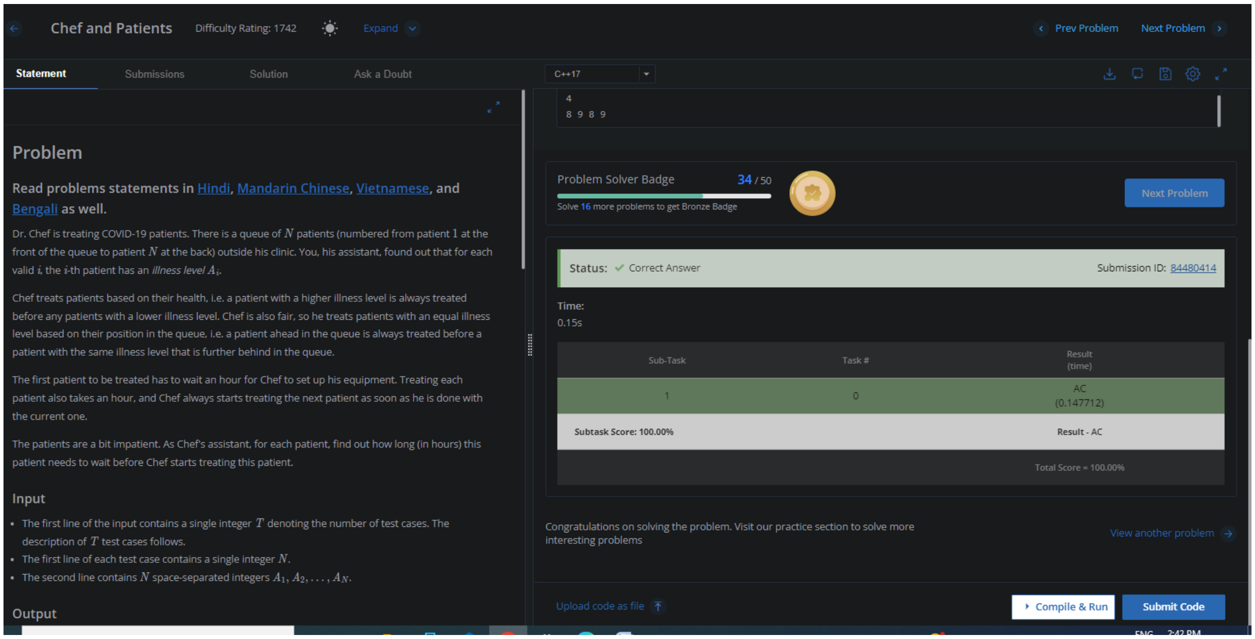
Congratulations on solving the problem. Visit our practice section to solve more interesting problems

Q5. Chef and Patients

Code:

```
#include <bits/stdc++.h>
using namespace std;


void solve() {
    int n ; cin >> n;
    vector<pair<int, int>> a(n);
    int mx = 0;
    for(int i=0; i<n; i++) {
        a[i].second = i;
        cin >> a[i].first;
        mx = max(mx, a[i].first);
    }
    for(int i=0; i<n; i++) a[i].first = mx - a[i].first;

    sort(a.begin(), a.end());
    vector<int> ans(n);

    for(int i=0; i<n; i++) ans[a[i].second] = i+1;

    for(int& e : ans) cout << e << " "; cout << endl;
}

int main() {
    int t; cin >> t;
    while(t--) solve();
            return 0;
}
```

OUTPUT



Problem

Read problems statements in Hindi, Mandarin Chinese, Vietnamese, and Bengali as well.

Dr. Chef is treating COVID-19 patients. There is a queue of N patients (numbered from patient 1 at the front of the queue to patient N at the back) outside his clinic. You, his assistant, found out that for each valid i, the i-th patient has an *illness level* A_i.

Chef treats patients based on their health, i.e. a patient with a higher illness level is always treated before any patients with a lower illness level. Chef is also fair, so he treats patients with an equal illness level based on their position in the queue, i.e. a patient ahead in the queue is always treated before a patient with the same illness level that is further behind in the queue.

The first patient to be treated has to wait an hour for Chef to set up his equipment. Treating each patient also takes an hour, and Chef always starts treating the next patient as soon as he is done with the current one.

The patients are a bit impatient. As Chef's assistant, for each patient, find out how long (in hours) this patient needs to wait before Chef starts treating this patient.

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first line of each test case contains a single integer N.
- The second line contains N space-separated integers $A_1, A_2, \ldots, A_N$.

Output

```
4
8 9 8 9
```

Problem Solver Badge 34 / 50

Solve 16 more problems to get Bronze Badge

Status: ✓ Correct Answer Submission ID: 84480414

Time:
0.15s

Sub-Task	Task #	Result (time)
1	0	AC (0.147712)
Subtask Score: 100.00%		**Result - AC**
		Total Score = 100.00%

Congratulations on solving the problem. Visit our practice section to solve more interesting problems