Worksheet 5

Student Name: Ravindra Singh Mehta

UID: 20BCS7060

Branch: CSE

Section/Group: DWWC-43

Semester: 5

Date of Performance: 09-01-2023

Subject Name: Data Structure

1. Aim/Overview of the practical:

i) Chef is given a sequence of integers A_1, A_2, \ldots, A_N$A_1, A_2, \ldots, A_N$.

Chef considers a contiguous subsequence A_l, A_{l+1}, \ldots, A_r$A_l, A_{l+1}, \ldots, A_r$ (where 1 \le l \le r \le N1$\le l \le r \le N$) *fruitful* if it satisfies the condition |A_l - A_r| = \mathrm{max}(A_l, A_{l+1}, \ldots, A_r) - \mathrm{min}(A_l, A_{l+1}, \ldots, A_r) \,.$|A_l - A_r| = \max(A_l, A_{l+1}, \ldots, A_r) - \min(A_l, A_{l+1}, \ldots, A_r)$.

Please help Chef find the number of fruitful contiguous subsequences of the sequence AA.

ii) You are given an array AA with length NN. On each day, the following process was performed:

- a new array RR is created; this array contains all elements A_iA_i such that i=1$i=1$, i=N$i=N$ or \mathop{\rm min}(A_{i-1}, A_{i+1}) \le A_imin$(A_{i-1}, A_{i+1}) \le A_i$ (2 \le i \lt N2$\le i < N$), in the original order
- AA is replaced by RR and NN is replaced by the length of RR

In other words, all elements of AA that are between two bigger elements disappear.

For each element of the original array, calculate the number of the day on which it disappeared (the process starts with day 11), or determine that it never disappeared.

iii) Hussain is very bored, and has a lot of exams lined up next week. But he doesn't want to study. As usual, he decided to fail his exams and play with Hasan instead (who has already failed). Hussain invented this new game to play with Hasan.

Hussain shows Hasan a multiset of integers. (A multiset is a collection of elements where there can be duplicates). In every move Hussain removes a maximum number from this multiset and divides it by 2 (integer division, which is rounded down).

If after dividing, the number is still positive (greater than 0) he re-inserts it back into the multiset.

Before the start of the game, Hussain shows Hasan the numbers in his multiset.

Hussain asks Hasan, **M** questions. The **i-th** question will be denoted by **Q[i]**, and Hasan must find the value of the number Hussain will be dividing (before he performs the division) after **Q[i]**-1 moves. That is, what is the value of the number on which the **Q[i]**-th division is performed?

Can you help Hasan and answer the queries?

iv) Harry is a bright student. To prepare thoroughly for exams, he completes all the exercises in his book! Now that the exams are approaching fast, he is doing book exercises day and night. He writes down and keeps updating the remaining number of exercises on the back cover of each book.

Harry has a lot of books messed on the floor. Therefore, he wants to pile up the books **that still have some remaining exercises** into a single pile. He will grab the books one-by-one and add the books that still have remaining exercises to the top of the pile.

Whenever he wants to do a book exercise, he will pick the book with the minimum number of remaining exercises from the pile. In order to pick the book, he has to remove all the books above it. Therefore, if there are more than one books with the minimum number of remaining exercises, he will take the one which requires the least number of books to remove. The removed books are returned to the messy floor. After he picks the book, he will do all the remaining exercises and trash the book.

Since number of books is rather large, he needs your help to tell him the number of books he must remove, for picking the book with the minimum number of exercises.

Note that more than one book can have the same name.

v) Dr. Phil has put you in charge of his new hospital while he is chilling at the ranch. N people have booked appointment with the Doctor today numbered from 11 to N such that person X booked appointment before person Y if $X < Y$.

Now, the people do not necessarily arrive at the hospital on time, so you decide of an unusual way of handling the situation. At the start you fulfill the appointment of the people who booked first (so if person 1 is available then that person can meet the doctor) otherwise you look for next available person.

If you fulfill the appointment of any person and their exist some people (1 or more) who have booked earlier but haven't arrived at the hospital yet, then those people are said to have missed their appointment.

If at some point of time there exists a set of people that have missed their appointment earlier and the person who booked appointment most recently among them (later than others in that set) say person X is now available then X is given the priority to meet the doctor over any other person. But even if any person in that set Y other than X is available and X is not available then you start appointments of the people who have not yet missed their appointments in their order of booking.

You are given the order in which people arrive at the hospital and have to print the order of their appointments.

2. **Steps for experiment/practical/Code:**

i)

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

using ll = long long;

void init(int bit[], int n)
{
    int i;
    for(i=0;i<n;i++)
        bit[i] = 0;
}
void update(int x, int bit[], int n, int val)
{
    while(x <= n)
    {
        bit[x]+=val;
        x+=(x&-x);
    }
}
int query(int x, int bit[])
{
    int ans = 0;
    while(x > 0)
    {
        ans+=bit[x];
        x-=(x&-x);
    }
    return ans;
}
ll solve(int arr[], int n)
{
```

```
    int bit[n+5];
    init(bit, n+5);
    int i;
    vector<int> s;
    int prev[n+1];
    for(i=1;i<=n;i++)
    {
        while(!s.empty() && arr[s.back()] <= arr[i])
            s.pop_back();
        if(s.empty())
            prev[i] = 0;
        else
            prev[i] = s.back();
        s.push_back(i);
    }
    s.clear();
    ll ans = 0;
    for(i=1;i<=n;i++)
    {
        while(!s.empty() && arr[s.back()] > arr[i])
        {
            update(s.back(), bit, n, -1);
            s.pop_back();
        }
        s.push_back(i);
        update(i, bit, n, 1);
        ans += (query(i, bit) - query(prev[i], bit));
    }
    return ans;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    int t;
    cin >> t;
```

```
    while(t--)
    {
        int n;
        cin >> n;
        int arr[n+1];
        int i;
        for(i=1;i<=n;i++)
            cin >> arr[i];
        ll ans1 = solve(arr, n);

        reverse(arr+1, arr+1+n);
        ans1 += solve(arr, n);

        reverse(arr+1, arr+n+1);
        for(i=1;i<=n;i++)
        {
            int j = i;
            while(j <= n && arr[j] == arr[i])
                j++;
            ll len = j - i;
            ans1 -= (len*(len+1)/2);
            i = j-1;
        }
        cout << ans1 << '\n';
    }
}
```

ii)

```
#include "bits/stdc++.h"
using namespace std;
void solve() {
	int n;
	cin >> n;
	vector<int> t(n, 0);
	stack<tuple<int, int, int>> st;

	for(int i = 0; i < n; ++i) {
		int a, end_time = 0;
		cin >> a;

		while(st.size() >= 2) {
			auto x = st.top();
			st.pop();
			auto y = st.top();
			if(get<0>(x) < min(get<0>(y), a)) t[get<2>(x)] = end_time = 1 + max(get<1>(x),
end_time);
			else {
				st.push(x);
				break;
			}
		}

		st.push({a, end_time, i});
	}

	for(int T : t) cout << T << ' ';
	cout << '\n';
}

int main() {
```

```
    ios_base :: sync_with_stdio(false);
    cin.tie(0);
    int t;
    cin >> t;
    while(t--) solve();
    return 0;
}
```

iii)

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

typedef long long ll;

int main(){
    int n,m;
    cin>>n>>m;

    ll arr[n];
    for(int i=0; i<n; i++)
    cin>>arr[i];
    sort(arr,arr+n);
    queue<ll> q;

    int end_p=n-1;
    int count_m=0;

   while(m--){
      int curr_m;
      cin>>curr_m;
      ll ans;

      for(;count_m<curr_m; count_m++){
    if(end_p>=0 && (q.empty() || (arr[end_p]>=q.front()))){
            ans=arr[end_p];
            end_p--;
        }else{
           ans=q.front();
           q.pop();
         }
         q.push(ans/2);
      }
      cout<<ans<<endl;
   }

    return 0;}
```

iv)

```
#include <iostream>
using namespace std;
#include<bits/stdc++.h>
int main() {
        // your code goes here
        ios_base::sync_with_stdio(false);
    cin.tie(NULL);
        int n;
        cin>>n;
        stack<pair<pair<int,string>,int>> st1;
        for(int i=0;i<n;i++){
            int x;
            cin>>x;
            string str;
            if(x!=-1 ){
                cin>>str;
            }else if(x==-1){
               cout<<st1.top().second<<" "<<st1.top().first.second<<endl;
                st1.pop();
                continue;
            }
            if (x==0){
                continue;
            }
            if(!st1.empty() and st1.top().first.first<x ){
               st1.top().second++;
               continue;
           }

               pair<int,string> p = make_pair(x,str);
               pair<pair<int,string>,int> whole_pair= make_pair(p,0);
               st1.push(whole_pair);
        }
        return 0;
}
```

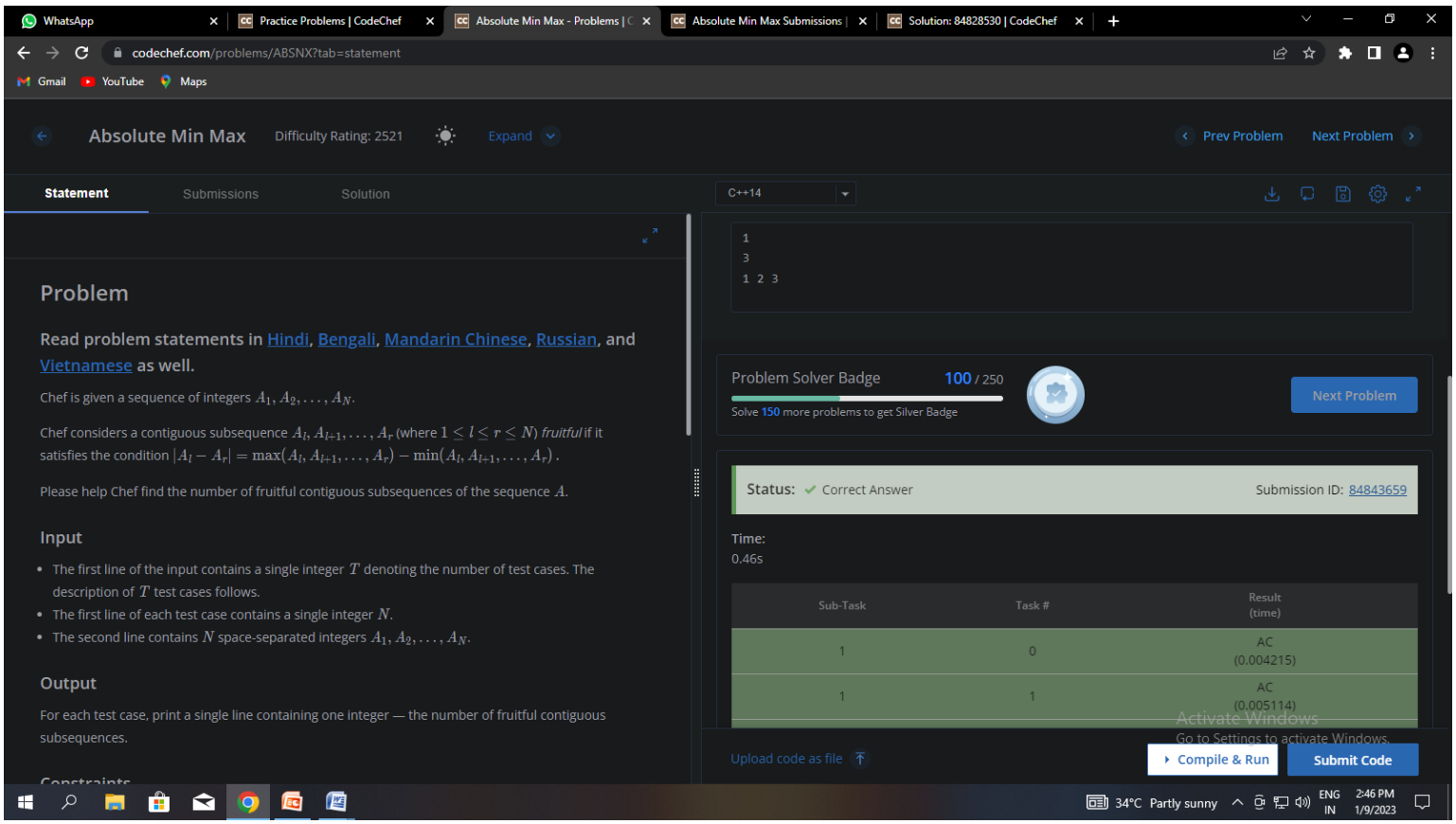
v)

```
#include <bits/stdc++.h>

using namespace std;

const int MAX = (int) 1e6 + 1;
const int N = (int) ('Z' - 'A') + 1;
#define rep(i, a, b) for (int i = a; i < b; ++i)
#define ll long long int
#define PII pair<int, int>
#define MP make_pair
#define PB push_back


void solve() {
	int n;
	scanf("%d", &n);

	int a[n + 1];
	queue<int> q;
	rep(i, 1, n + 1) {
	   scanf("%d", &a[i]);
	   q.push(i);
	}

	vector<bool> met(n + 1, false);
	vector<bool> available(n + 1, false);
	stack<int> st;

	rep(i, 1, n + 1) {
		available[a[i]] = true;
		if (!q.empty() && a[i] == q.front()) {
			cout << a[i] << " ";
			q.pop();
		}
		else if (!q.empty()) {
```

```
                int t = q.front();

                while (!q.empty() && q.front() < a[i]) {
                    int top = q.front();
                        st.push(top);
                        q.pop();
                }
                if (a[i] > t) {
                        cout << a[i] << " ";
                        q.pop();
                }
            }
            while (!st.empty() && available[st.top()]) {
                cout << st.top() << " ";
                st.pop();
            }
        }

}
int main()
{
        ios_base::sync_with_stdio(0);
        cin.tie(0);
        int t;
        scanf("%d", &t);
        for (int it = 1; it <= t; ++it) {
            //cout << "Case #" << it << ": ";
            solve();
            cout << "\n";
        }

        return 0;
}
```

3. Observations/Discussions/ Complexity Analysis:

i)



ii)

WhatsApp | Practice Problems | CodeChef | Weak in the Middle - Problems | | Weak in the Middle Submissions | Solution: 84843241 | CodeChef

codechef.com/problems/WEAKMID?tab=statement

Gmail YouTube Maps

Weak in the Middle Difficulty Rating: 2476 Expand Prev Problem Next Problem

Statement Submissions Solution C++14

Problem

Read problems statements in Mandarin chinese and Vietnamese as well.

You are given an array A with length N. On each day, the following process was performed:

- a new array R is created; this array contains all elements A_i such that $i = 1$, $i = N$ or $\min(A_{i-1}, A_{i+1}) \le A_i$ $(2 \le i < N)$, in the original order
- A is replaced by R and N is replaced by the length of R

In other words, all elements of A that are between two bigger elements disappear.

For each element of the original array, calculate the number of the day on which it disappeared (the process starts with day 1), or determine that it never disappeared.

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 N space-separated integers. For each valid i, the i-th of these integers should be the number of the day on which the i-th element of the original array

```
1
6
3 2 5 4 1 7
```

Problem Solver Badge 101 / 250

Solve 149 more problems to get Silver Badge

Next Problem

Status: ✔ Correct Answer Submission ID: 84843983

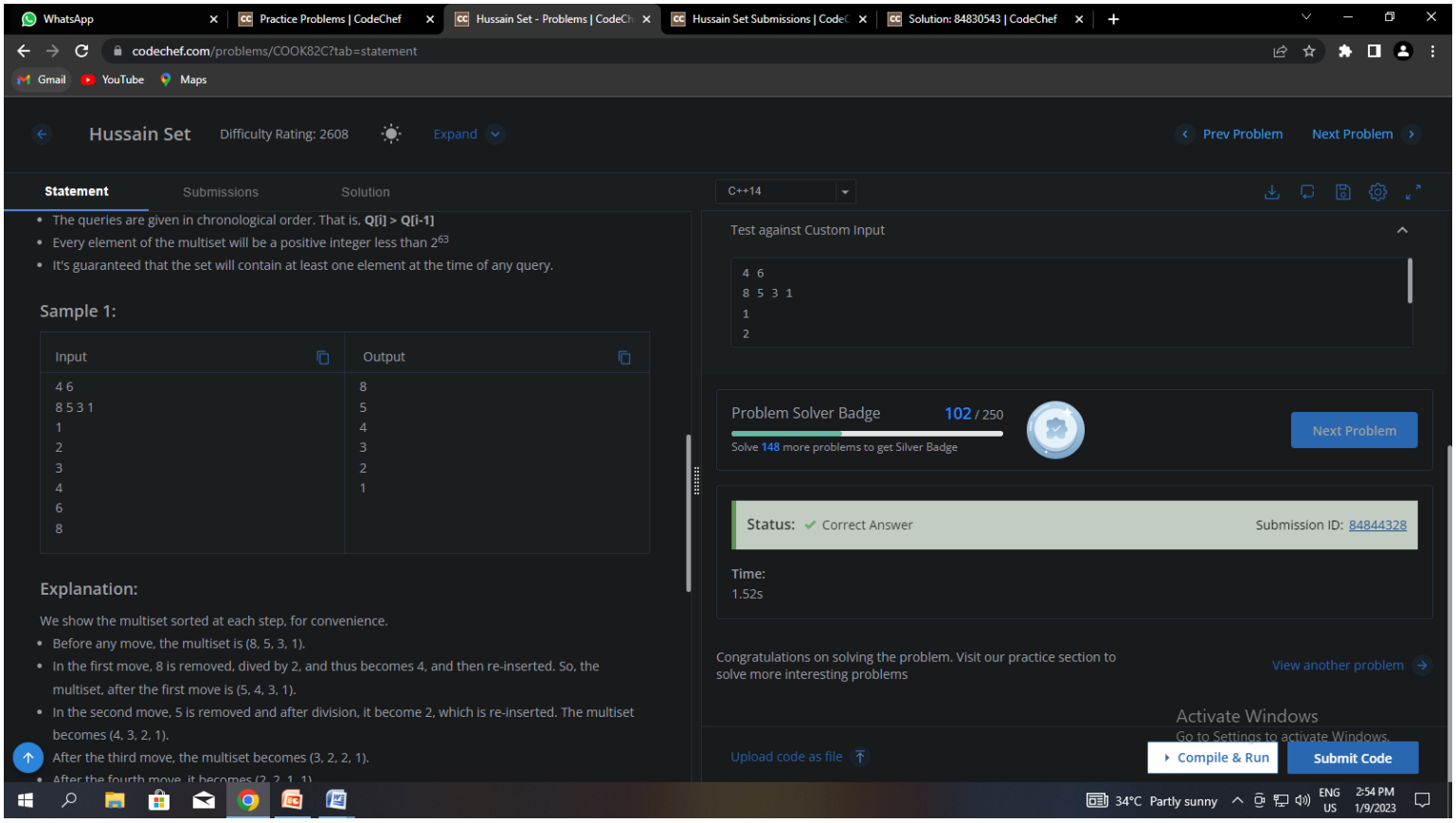
Time:
0.02s

Sub-Task	Task #	Result (time)
1	0	AC (0.003756)
1	1	AC (0.003785)

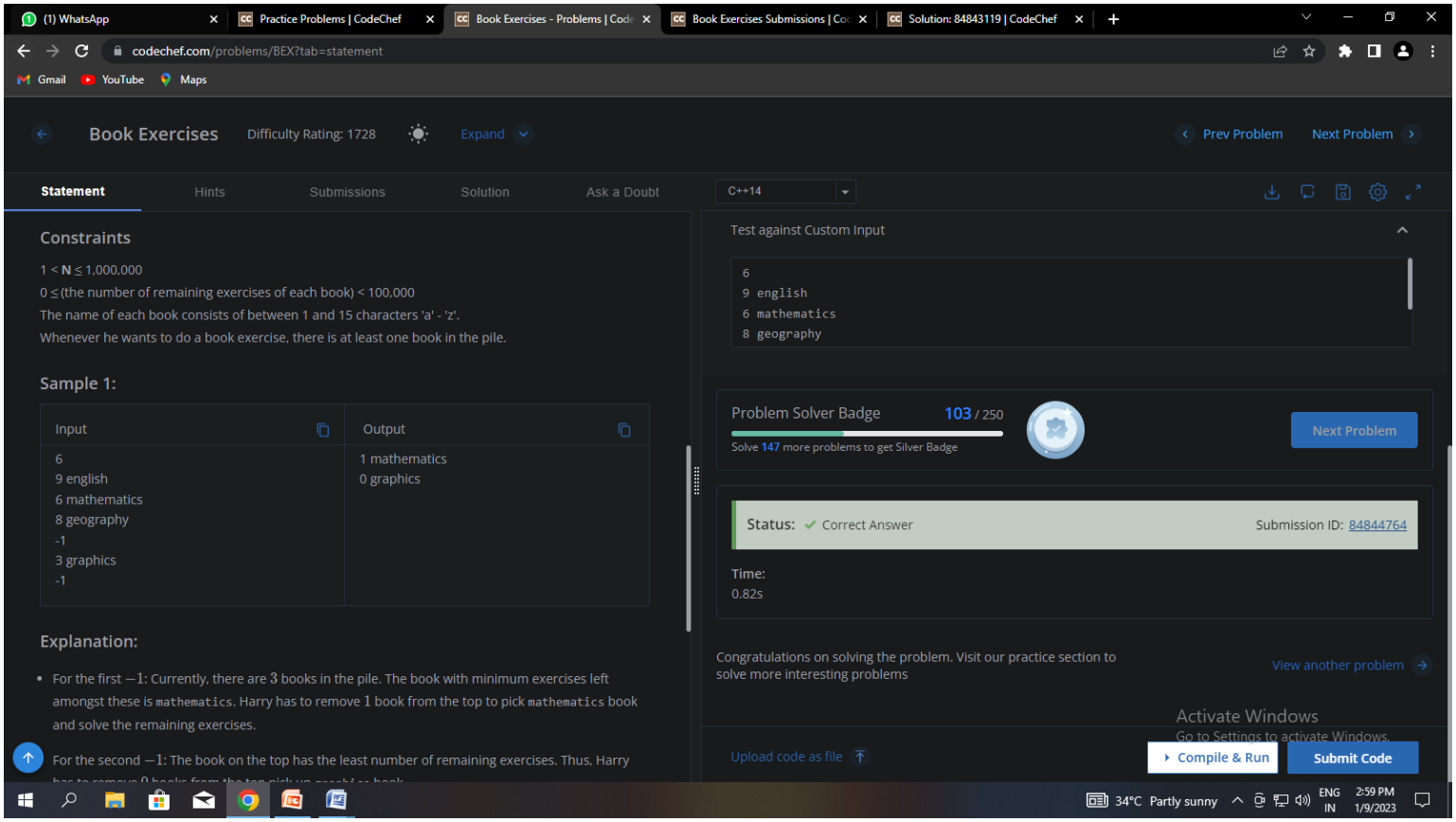
Upload code as file Compile & Run Submit Code

34°C Partly sunny ENG US 2:49 PM 1/9/2023

iii)


WhatsApp
Practice Problems | CodeChef
Hussain Set - Problems | CodeCh
Hussain Set Submissions | Code
Solution: 84830543 | CodeChef
codechef.com/problems/COOK82C?tab=statement
Gmail
YouTube
Maps
Hussain Set
Difficulty Rating: 2608
Expand
Prev Problem
Next Problem
Statement
Submissions
Solution
C++14
The queries are given in chronological order. That is, Q[i] > Q[i-1]
Every element of the multiset will be a positive integer less than 2^63
It's guaranteed that the set will contain at least one element at the time of any query.
Sample 1:
Input
4 6
8 5 3 1
1
2
3
4
6
8
Output
8
5
4
3
2
1
Explanation:
We show the multiset sorted at each step, for convenience.
Before any move, the multiset is (8, 5, 3, 1).
In the first move, 8 is removed, dived by 2, and thus becomes 4, and then re-inserted. So, the multiset, after the first move is (5, 4, 3, 1).
In the second move, 5 is removed and after division, it become 2, which is re-inserted. The multiset becomes (4, 3, 2, 1).
After the third move, the multiset becomes (3, 2, 2, 1).
Test against Custom Input
4 6
8 5 3 1
1
2
Problem Solver Badge
102 / 250
Solve 148 more problems to get Silver Badge
Next Problem
Status: Correct Answer
Submission ID: 84844328
Time:
1.52s
Congratulations on solving the problem. Visit our practice section to solve more interesting problems
View another problem
Activate Windows
Go to Settings to activate Windows.
Upload code as file
Compile & Run
Submit Code
34°C Partly sunny
ENG US
2:54 PM
1/9/2023

iv)


(1) WhatsApp
Practice Problems | CodeChef
Book Exercises - Problems | Code
Book Exercises Submissions | Co
Solution: 84843119 | CodeChef
codechef.com/problems/BEX?tab=statement
Gmail
YouTube
Maps
Book Exercises
Difficulty Rating: 1728
Expand
Prev Problem
Next Problem
Statement
Hints
Submissions
Solution
Ask a Doubt
C++14
Constraints
1 < N ≤ 1,000,000
0 ≤ (the number of remaining exercises of each book) < 100,000
The name of each book consists of between 1 and 15 characters 'a' - 'z'.
Whenever he wants to do a book exercise, there is at least one book in the pile.
Sample 1:
Input
6
9 english
6 mathematics
8 geography
-1
3 graphics
-1
Output
1 mathematics
0 graphics
Explanation:
For the first −1: Currently, there are 3 books in the pile. The book with minimum exercises left amongst these is mathematics. Harry has to remove 1 book from the top to pick mathematics book and solve the remaining exercises.
For the second −1: The book on the top has the least number of remaining exercises. Thus, Harry
Test against Custom Input
6
9 english
6 mathematics
8 geography
Problem Solver Badge
103 / 250
Solve 147 more problems to get Silver Badge
Next Problem
Status: Correct Answer
Submission ID: 84844764
Time:
0.82s
Congratulations on solving the problem. Visit our practice section to solve more interesting problems
View another problem
Activate Windows
Go to Settings to activate Windows.
Upload code as file
Compile & Run
Submit Code
34°C Partly sunny
ENG IN
2:59 PM
1/9/2023

v)

(1) WhatsApp | Practice Problems | CodeChef | Dr Phil goes to the ranch - Probl | Solution: 84829884 | CodeChef

codechef.com/problems/CAC202?tab=statement

Gmail YouTube Maps

Dr Phil goes to the ranch

Difficulty Rating: NA

Expand

Statement Submissions Solution

Problem

Dr. Phil has put you in charge of his new hospital while he is chilling at the ranch. N people have booked appointment with the Doctor today numbered from 1 to N such that person X booked appointment before person Y if $X < Y$.

Now, the people do not necessarily arrive at the hospital on time, so you decide of an unusual way of handling the situation. At the start you fulfill the appointment of the people who booked first (so if person 1 is available then that person can meet the doctor) otherwise you look for next available person.

If you fulfill the appointment of any person and their exist some people (1 or more) who have booked earlier but haven't arrived at the hospital yet, then those people are said to have missed their appointment.

If at some point of time there exists a set of people that have missed their appointment earlier and the person who booked appointment most recently among them (later than others in that set) say person X is now available then X is given the priority to meet the doctor over any other person. But even if any person in that set Y other than X is available and X is not available then you start appointments of the people who have not yet missed their appointments in their order of booking.

You are given the order in which people arrive at the hospital and have to print the order of their appointments.

$Note$: Use fast input/output methods. Python users should submit their solutions in PYPY 3.

C++14

```
17
18      int a[n + 1];
19      queue<int> q;
20      rep(i, 1, n + 1) {
21         scanf("%d", &a[i]);
22         q.push(i);
```

67:1

Test against Custom Input

Your program will run with no input

Status : Successfully executed

Time: 0.007885 secs

Memory: 5.352 Mb

No details to show

Upload code as file

Compile & Run

Submit Code

Activate Windows
Go to Settings to activate Windows.

34°C Partly sunny ENG IN 3:02 PM 1/9/2023

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			