Course: ESO207A – Data Structures and Algorithms

Indian Institute of Technology Kanpur

Programming Assignment 2 : *Data Structures !!*

Important guidelines for submission

- It is only through the assignments that one learns the most about the algorithms and data structures. You are advised to refrain from searching for a solution on the net or from a notebook.
- Refrain from collaborating with the students of other groups or your friends. If any evidence is found that confirms copying, the penalty will be very harsh. Refer to the website at the link: https://cse.iitk.ac.in/pages/AntiCheatingPolicy.html regarding the departmental policy on cheating.
- **The assignment is to be submitted on Gradescope. You must submit one C file named as *assign2.c* for each question. This is EXTREMELY IMPORTANT. The code is checked with an autograder, and a different file name will fail to pass the test cases.**
- **DEADLINE** for this assignment submission is 11 PM Friday (8th Feb).
- In case of any issue related to this assignment, send an email at abhijeeta21@iitk.ac.in or anuj21@iitk.ac.in
- **Note**: Write the output functions for question1, question2, question3 and question4 as function1(), function2(), function3(), function4() respectively. Refer the sample code for more details.

Tasks to be done

1 Maximum in a queue

1.1 Problem Statement

Suppose you are working in a famous shop where there is always a long queue of buyers waiting to purchase what they want. Each buyer i has a value $v[i]$ associated with it that represents the amount of things he is going to purchase. Being experienced enough, you can tell the value of a customer just by looking at their carts.
The shop owner however is really fond of data, so he randomly keeps asking you what is the maximum value of any buyer in the queue. You need to somehow answer his questions very quickly otherwise your job is in danger!
Formally there are 3 events possible, let's call them event 1, 2, 3 as follows:

1. Event 1: A new person with value v joins the queue (at the last position)
2. Event 2: The first person in queue completes his purchase and exits the shop
3. Event 3: The shop owner asks you what is the current maximum value in the queue.

Your task is to process all these events in order they are provided and output the answer to all type 3 events. The expected time complexity of your solution is $O(q)$ where q is the number of events.

1.2 Input Format

- The first line contains an integer q - the total number of events.
- q lines follow, each line can be of 3 types
  - `1 v`
    A new person with value v gets in the queue (v is always a positive integer)
  - `2`
    The first person in the queue exits
  - `3`
    You need to report the current maximum value in the queue

1.3 Output Format

Output the answer to all type 3 queries in separate lines.

1.4 Constraints

- $2 \leq q \leq 10^6$
- $1 \leq v[i] \leq 10^9 \ \forall i$
- You can assume that there won't be any type 2 or 3 event if the queue is empty at any stage.

1.5 Example

Input

```
8
1 5
1 3
3
1 4
2
3
2
3
```

Output

```
5
4
4
```

Explanation Initially buyers with value 5 and 3 are added to queue so maximum is 5, then 4 is added and 5 leaves, so maximum is 4, then 3 also leaves and the maximum is again 4.

1.6 Hints

- *Note that Event 1 and 2 are regular operations on a queue and can easily be done in O(1) time, the challenge is to answer the type 3 queries. Try to think of how you can maintain the maximum of the queue with some additional data structure.*
- *Recall that you have studied a data structure in class that supports arbitrary insertion, deletion and reporting the maximum, but that will take $O(log(n))$ time for these operations and would be quite complex to implement.*
- *Observe that here the insertion and deletion are not arbitrary, they are operations that you can perform on a queue. Try to take advantage of this restriction.*

2 Range Minima

2.1 Problem Statement

You need to implement the range-minima data structure that you have studied in class. You will be given an array of n integers and q queries. Each query will compose of 2 integers i and j and you need to output the minimum value in the subarray $a[i], \ldots, a[j]$. The expected time complexity of each query is $O(1)$ and the preprocessing time for creating the data structure should be $O(n \log n)$.

2.2 Input Format

- The first line contains two integer n and q - the number of elements in the array and the number of queries.
- The second line contains n integers (space separated) - The elements of array a
- The next q lines contains 2 integers i and j (1-based indexing).

2.3 Output Format

Output q lines, each line containing a single integer - the answer of the corresponding range minima query.

2.4 Constraints

- $2 \leq n \leq 10^5$
- $1 \leq q \leq 10^5$
- $-10^9 \leq a[i] \leq 10^9 \ \forall i$
- For each query, $1 \leq i \leq j \leq n$

2.5 Example

Input

```
7 5
4 8 6 2 7 5 4
1 4
2 3
3 6
1 7
5 7
```

Output

```
2
6
2
2
4
```

Explanation
Given array $a = [4, 8, 6, 2, 7, 5, 4]$

- 1st query: $i = 1,\ j = 4 : min(4, 8, 6, 2) = 2$
- 2nd query: $i = 2,\ j = 3 : min(8, 6) = 6$
- 3rd query: $i = 6,\ j = 6 : min(6, 2, 7, 5) = 2$
- 4th query: $i = 1,\ j = 7 : min(4, 8, 6, 2, 7, 5, 4) = 2$
- 5th query: $i = 5,\ j = 7 : min(7, 5, 4) = 4$

2.6 Hints

- *Discussed in lectures*

3 Number of Binary Search Trees

3.1 Problem Statement

Binary Search Tree is a very useful data structure, however, have you ever wondered how many different structures of binary search trees are possible for n distinct values? Two binary search trees are different if they are structurally different at atleast one place (equivalently two binary search trees are considered same if all the left and right children of all the corresponding nodes are same in both the trees).
Given n you need to output the number of binary search trees containing n distinct nodes. Since the answer can be really huge, just output its remainder modulo $10^9 + 7$

3.2 Input Format

- Each test case contains a single integer n - the total number of nodes in binary search tree.

3.3 Output Format

Output the number of distinct binary search trees possible.

3.4 Constraints

- $1 \leq n \leq 1000$

3.5 Example

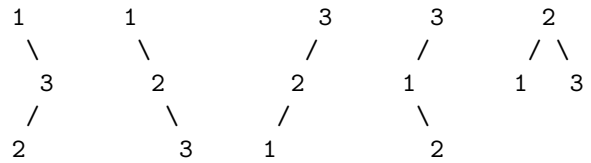
Input

```
3
```

Output

```
5
```

Explanation
Let say node values are 1, 2 and 3.
5 possible trees are:



3.6 Hints

- *First try to observe some recurrent pattern, you can implement it as a recursive function, it will give you 60% of the marks*
- *Next try to observe some redundancy in your recursive function, are there values that you are computing many times during the recursive call? Think in the direction that if you store those values in an array to avoid the recomputation every time, it can significantly improve the time taken for the algorithm*
- *Optionally, if you are interested, try to formally analyze, how this optimization affects the time complexity of your algorithm*

4 k-score

4.1 Problem Statement

Alice is a budding programmer who has recently been introduced to arrays. One day, her mentor gave her an interesting task involving arrays and queries, but she's struggling to solve it efficiently. She's now asking for your help!
Alice is given an array a of size n and needs to answer q queries. Each query provides an integer k, and for each query, she must determine the k-score of the array. The k-score of an array is defined as the

length of the longest contiguous subarray of a such that the minimum value in that subarray is equal to k. More formally, the k score of an array is defined as:

$$k\text{-score}(a, k) = \max_{1 \le i \le j \le n} \big(j - i + 1\big)$$

subject to the condition:

$$\min(a[i], a[i+1], \ldots, a[j]) = k,$$

where $1 \le i \le j \le n$. Design an O(n) algorithm to find k-score for all the queries.

4.2 Input Format

The first line contains two integers n and q - the size of the array a and the number of queries, respectively. The second line contains n integers, the elements of the array a. The next q lines contain a single integer k, representing a query.

4.3 Output Format

For each query, output a single integer,the k-score of the array for the given k.

4.4 Constraints

- $1 \le n \le 10^5$
- $1 \le q \le 10^5$
- $1 \le a[i] \le 10^5 \quad \text{for } 1 \le i \le n$
- $1 \le k \le 10^9$

4.5 Example

Input

```
5 4
1 3 2 4 2
2
3
1
8
```

Output

```
4
1
5
0
```

Explanation

```
Longest subarray with minimum element equal to 2:
    3 2 4 2
Longest subarray with minimum element equal to 3:
    3
Longest subarray with minimum element equal to 1:
    1 3 2 4 2
Longest subarray with minimum element equal to 8:
Since 8 is not present in the array , therefore there does not exist any subarray with 8 as
the minimum element.
```

4.6 Hints:

- *What would be the k-score if element k is not present in the array?*
- *If we know the index of previous smaller element for element k, can we conclude anything about the longest length subarray with minimum value k?*
- *Can we somehow pre-compute the k-score for every element k in the array?*

Test Cases

You will be evaluated on several hidden test cases. These test cases will not be revealed to you. For each hidden test case, you need to match the expected output. Then only will you get the full marks. You will not see your score until the deadline is over when we grade your code using autograder, and publish the result. You may resubmit as many times as you want, but only the final submission will be graded.