Array and Queries ☆

247.86 more points to get your next star!

Rank: **280189** | Points: **227.14/475**

Problem Solving ★★★

Problem Submissions Leaderboard

Given an array, you are asked to perform a number of queries and divide the array into what are called, beautiful subsequences.

The array A has length n. A function $f(A)$ is defined to be a minimal possible x, such that it's possible to divide array A into x beautiful subsequences. Note that each element of an array should belong to exactly one subsequence, and subsequence does not necessarily need to be consecutive.

A subsequence S with length len is called beautiful if and only if:

- $len = 1$ or
- Let S' be a sorted version of S. It must hold that $S'_i = S'_{i+1} - 1$ for every $i \in [1, len - 1]$.

For instance, if $A = [1, 2, 3, 4, 3, 5]$, $f(A)$ would be 2. Because, you can divide A into 2 beautiful subsequences either like $[1, 2, 3]$ and $[4, 3, 5]$ or like $[1, 2, 3, 4, 5]$ and $[3]$.

You have to answer q queries. Each query is of the type:

- $id\ val$: you need to change a value of A_{id} to val, i.e. $A_{id} = val$. Here id is $1 - indexed$.

After each query, for the value of $f(A)$, lets denote that value as ans_i, where i indicates the i^{th} query.

You need to find $\sum_{i=1}^{q} i \times ans_i$ modulo $(10^9 + 7)$.

Input Format

The first line contains a single integer n, representing the length of array A.

The next line contains the array A given as space-separated integers.

The next line contains a single integer q, representing the number of queries.

Each of the q lines contain two integers id and val, which is described above.

Constraints

- $1 \leq n, q \leq 3 \times 10^5$
- $1 \leq A_i \leq 10^9$
- $1 \leq id \leq n$
- $1 \leq val \leq 10^9$

Output Format

Print the required answer in one line.

Sample Input 0

```
5
2 2 1 1 1
2

3 2
```

```
5 5
```

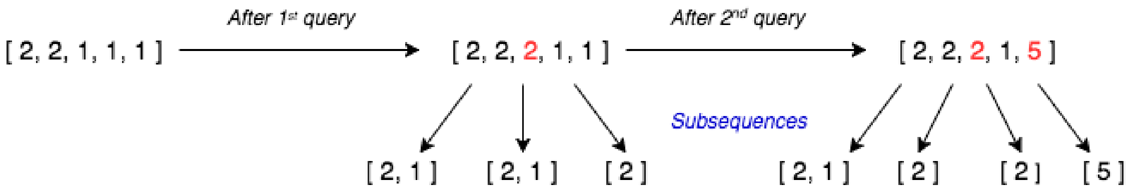
Sample Output 0

```
11
```

Explanation 0

The initial array A is $[2, 2, 1, 1, 1]$

- After 1^{st} query the array becomes $[2, 2, 2, 1, 1]$ this can be divided into 3 subsequences as $[2, 1]$, $[2, 1]$ and $[2]$.
- After 2^{nd} query the array becomes $[2, 2, 2, 1, 5]$ this can be divided into 4 subsequences as $[2, 1]$, $[2]$, $[2]$ and $[5]$.



Hence, calculating $\sum i \times ans_i$ we get

$$1 \times 3 + 2 \times 4 \Rightarrow 11$$

Sample Input 1

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

Sample Output 1

```
9
```

Explanation 1

The initial array A is $[3, 3]$

- After 1^{st} query the array becomes $[3, 4]$ this can be divided into 1 subsequence as $[3, 4]$.
- After 2^{nd} query the array becomes $[5, 4]$ this can be divided into 1 subsequence as $[5, 4]$.
- After 3^{rd} query the array becomes $[5, 2]$ this can be divided into 2 subsequences as $[5]$ and $[2]$.

Hence, calculating $\sum i \times ans_i$ we get

$$1 \times 1 + 2 \times 1 + 3 \times 2 \Rightarrow 9$$

Current Buffer (saved locally, editable)

Java 7

Java 7

```java
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.regex.*;

public class Solution {

    // Complete the arrayAndQueries function below.
    static int arrayAndQueries(int[] A, int[][] queries) {


    }

    private static final Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) throws IOException {
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT_PATH")));

        int n = scanner.nextInt();
        scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

        int[] A = new int[n];

        String[] AItems = scanner.nextLine().split(" ");
        scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

        for (int i = 0; i < n; i++) {
            int AItem = Integer.parseInt(AItems[i]);
            A[i] = AItem;
        }

        int q = scanner.nextInt();
        scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

        int[][] queries = new int[q][2];

        for (int i = 0; i < q; i++) {
            String[] queriesRowItems = scanner.nextLine().split(" ");
            scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

            for (int j = 0; j < 2; j++) {
                int queriesItem = Integer.parseInt(queriesRowItems[j]);
                queries[i][j] = queriesItem;
            }
        }

        int result = arrayAndQueries(A, queries);

        bufferedWriter.write(String.valueOf(result));
        bufferedWriter.newLine();

        bufferedWriter.close();

        scanner.close();
    }
}

```

Line: 1 Col: 1

Upload Code as File

□ Test against custom input

Run Code

Submit Code