

[HOME](#) [TOP](#) [CONTESTS](#) [GYM](#) [PROBLEMSET](#) [GROUPS](#) [RATING](#) [API](#) [HELP](#) [KOTLIN HEROES](#)  [DASHA](#)  [CALENDAR](#)
[PROBLEMS](#) [SUBMIT CODE](#) [MY SUBMISSIONS](#) [STATUS](#) [HACKS](#) [ROOM](#) [STANDINGS](#) [CUSTOM INVOCATION](#)

G2. Into Blocks (hard version)

time limit per test: 5 seconds
 memory limit per test: 256 megabytes
 input: standard input
 output: standard output

This is a harder version of the problem. In this version $q \leq 200\,000$.

A sequence of integers is called *nice* if its elements are arranged in blocks like in $[3, 3, 3, 4, 1, 1]$. Formally, if two elements are equal, everything in between must also be equal.

Let's define *difficulty* of a sequence as a minimum possible number of elements to change to get a nice sequence. However, if you change at least one element of value x to value y , you must also change all other elements of value x into y as well. For example, for $[3, 3, 1, 3, 2, 1, 2]$ it isn't allowed to change first 1 to 3 and second 1 to 2. You need to leave 1's untouched or change them to the same value.

You are given a sequence of integers a_1, a_2, \dots, a_n and q updates.

Each update is of form " $i\ x$ " — change a_i to x . Updates are not independent (the change stays for the future).

Print the difficulty of the initial sequence and of the sequence after every update.

Input

The first line contains integers n and q ($1 \leq n \leq 200\,000$, $0 \leq q \leq 200\,000$), the length of the sequence and the number of the updates.

The second line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 200\,000$), the initial sequence.

Each of the following q lines contains integers i_t and x_t ($1 \leq i_t \leq n$, $1 \leq x_t \leq 200\,000$), the position and the new value for this position.

Output

Print $q + 1$ integers, the answer for the initial sequence and the answer after every update.

Example

| input | Copy |
|----------------------------------------------------|------|
| <pre> 5 6 1 2 1 2 1 2 1 4 1 5 3 2 3 4 2 2 1 </pre> | |
| output | Copy |
| <pre> 2 1 0 0 2 3 0 </pre> | |

Codeforces Round #584 - Dasha Code Championship - Elimination Round (rated, open for everyone, Div. 1 + Div. 2)

Finished

Practice



→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Practice

You are registered for practice. You can solve problems unofficially. Results can be found in the contest status and in the bottom of standings.

→ Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest

→ Submit?

Language: GNU G++11 5.1.0

Choose file: 选择文件 未选择任何文件

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.



Submit

→ Problem tags

data structures *3200

No tag edit access

→ Contest materials

- [Announcement \(en\)](#) 
- [Tutorial \(en\)](#) 

[Codeforces](#) (c) Copyright 2010-2019 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Sep/16/2019 15:07:53^{UTC+8} (e2).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Supported by



ITMO UNIVERSITY