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 $q \le 200\,000$.

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 $a_1, a_2, \ldots, a_n$ 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 ($1 \le n \le 200\,000$, $0 \le q \le 200\,000$), the length of the sequence and the number of the updates.

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

Each of the following q lines contains integers i_t and x_t ($1 \le i_t \le n$, $1 \le x_t \le 200\,000$), 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

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

output

```
2
1
0
0
2
3
0
```

- Announcement (en)
- Tutorial (en)