

Coloured Walkway

Input File	Output File	Time Limit	Memory Limit
standard input	standard output	2 seconds	1024 MiB

A new art installation has appeared in the streets of your local mall. It's a long walkway of n tiles, numbered 1 to n from left to right. Each tile is coloured one of c possible colours, numbered from 1 to c . The i -th tile has colour x_i .

You start on the leftmost tile and must get to the rightmost tile. You may only move by making jumps to the right. Using your powerful legs, you can skip any positive number of tiles with each jump.

Things are not so simple, however. The artist insists that you may only jump from one coloured tile to another colour if the colours are *compatible*! You have a list of p pairs of compatible colours (a colour may or may not be compatible with itself). The j -th such pair states that colours a_j and b_j are compatible.

The colours are very pretty and this compels you to count the number of ways you can get from the leftmost tile to the rightmost one by jumping. Two ways are considered different if there is a tile you land on in one way, but not the other. Since the answer can be quite big, give the answer modulo 1 000 000 007.

As this number may be large, you may wish to use 64-bit integer types (such as `long long` in C/++) in your program.

Subtasks and Constraints

For all subtasks:

- $2 \leq n \leq 100\,000$
- $1 \leq c \leq 100\,000$
- $1 \leq x_i \leq c$, for all i
- $0 \leq p \leq 100\,000$
- $1 \leq a_j, b_j \leq c$ for all j (Note that it is possible that $a_j = b_j$)
- $(a_i, b_i) \neq (a_j, b_j)$ and $(a_i, b_i) \neq (b_j, a_j)$ for any $i \neq j$. That is, no compatible pair of colours will appear twice.

Furthermore:

- For Subtask 1 (4 points): $c = 1$
- For Subtask 2 (22 points): $c \leq 100$.
- For Subtask 3 (15 points): $x_i = i$, for all i .
- For Subtask 4 (26 points): $p = c - 1$, $a_j = j$ and $b_j > j$ for all j .
- For Subtask 5 (33 points): No further constraints apply.

Input

The first line of input will contain the two integers n, c . The second line will contain the n integers x_1, x_2, \dots, x_n .

The third line contains the single integer p . Then, p lines follow, describing the pairs of compatible colours. The j -th such pair states that colours a_j and b_j are compatible.

Output

Output a single integer, the total number of ways of getting from the leftmost tile to the rightmost tile, modulo 1 000 000 007.

Sample Input 1

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

Sample Output 1

```
5
```

Sample Input 2

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

Sample Output 2

```
0
```

Sample Input 3

```
6 2
1 1 1 2 2 1
3
1 1
2 1
2 2
```

Sample Output 3

16

In the first sample case, there are 5 ways, which are listed below by the tile numbers they land on:

- $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$
- $1 \rightarrow 2 \rightarrow 5$
- $1 \rightarrow 4 \rightarrow 5$
- $1 \rightarrow 3 \rightarrow 4 \rightarrow 5$
- $1 \rightarrow 5$

In the second sample case, it is not possible to reach the rightmost tile, no matter how you jump! Hence the answer is 0.

In the third sample case, every colour is compatible with every other colour (including themselves), so any jump is allowed. There are 16 ways.