Have You Seen This Subarray?

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

There is an array a with $a[i] = i$ initially. m operations are performed with this array. During each operation, two **random** indexes i and j are chosen, and elements $a[i]$ and $a[j]$ are swapped.

You need to answer q queries. Each query is array b_1, b_2, ..., b_k. You need to find the first time when b was a continuous subarray of a.

Indexes in swap operations are guaranteed to be chosen independently from a uniform distribution.

Input

The first line contains three integers n, m, and q ($1 \le n, m, q \le 10^5$).

Each of the following m lines contains two integers i and j ($1 \le i < j \le n$) — descriptions of swap operations.

The next q lines contain descriptions of the queries. Each description starts with integer k ($1 \le k \le n$) and then k integers b_1, b_2, ..., b_k ($1 \le b_i \le n$). It is guaranteed that the sum of all k doesn't exceed 10^5.

Output

For each query described by the array b, you need to print the number of operations performed before b became a continuous subarray of a. If the initial array a contained b, print 0. It is guaranteed that b was a subarray of a at some point.

Example

standard input	standard output
6 3 5 1 5 3 4 1 6 2 4 1 3 3 1 5 3 3 4 5 4 5 2 4 3 2 6 2	1 3 0 2 3