Catch the Monster

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

A prehistoric monster has arrived on Earth via Dr. Peter's time machine. You need to help Dr. Peter catch it.

The monster hides in a forest with n vertices and m edges. Here, a forest is an acyclic undirected graph that may consist of multiple trees. You can perform the following operations to catch the monster:

- First, you choose a vertex x ($1 \le x \le n$).
- Then, if the monster is currently at vertex x, it is caught.
- Otherwise, the monster remains uncaught, and it may move to any adjacent vertex to its current vertex after the operation, except vertex x. Or it may choose not to move and stay at the same vertex.

We define a forest as *nice* if and only if there exists a finite sequence a of vertices, such that regardless of the monster's initial position and how it moves, performing operations by selecting vertices in the order of a can guarantee that the monster will be caught.

Now, you need to answer q questions from Dr. Peter. In each question, he gives you an interval $[l, r]$ ($1 \le l \le r \le n$). You need to tell him whether the subforest induced by the vertices whose indices are in $[l, r]$ (i.e., the graph formed by retaining only these vertices and the edges between them) is *nice*.

Input

The first line of the input contains three integers n, m, and q ($2 \le n \le 10^6$, $1 \le m \le n-1$, $1 \le q \le 10^6$), where n is the number of vertices in the forest, m in the number of edges in the forest, and q is the number of queries.

The next m lines of the input each contain two integers u and v ($1 \le u, v \le n$, $u \ne v$), representing an edge in the forest.

The next q lines of the input each contain two integers l and r ($1 \le l \le r \le n$), representing a query.

Output

For each query, output a single line "`Yes`" if the subforest is *nice*; otherwise, output a single line "`No`".

You can output the answer in any case (upper or lower). For example, the strings "`yEs`", "`yes`", "`Yes`", and "`YES`" will be recognized as positive responses.

Examples

standard input	standard output
10 9 3 1 2 1 3 1 8 2 5 2 6 2 7 3 4 8 9 8 10 1 3 2 6 1 10	Yes Yes No
100000 1 1 1 2 1 9999	Yes

Note

In the first test case:

- In the first query, for the subforest $[1, 3]$, you can set $a = [3, 1, 2]$.
- In the second query, for the subforest $[2, 6]$, you can set $a = [3, 4, 5, 2, 6]$.
- In the third query, for the subforest $[1, 10]$, it can be proven that there does not exist a finite valid sequence a.

In the second test case:

- In the only query, for the subforest $[1, 9999]$, you can set $a = [1, 2, \ldots, 9999]$.