



## Task 4: Curtains

Benson the Rabbit is organizing a performance on his plane!

He has a stage with  $n$  sections numbered 1 to  $n$  from left to right. He also has  $m$  curtains numbered from 1 to  $m$ .

Each of these  $m$  curtains can be lowered. Lowering curtain  $i$  covers sections  $l[i]$  to  $r[i]$ . A **curtain configuration** is a set of lowered curtains. Given a curtain configuration, a section  $x$  ( $1 \leq x \leq n$ ) is **covered** if and only if there exists a **lowered** curtain  $i$  such that  $l[i] \leq x \leq r[i]$ .

Benson wants to give a total of  $q$  performances, numbered from 1 to  $q$ . For each performance  $j$ , Benson requires a curtain configuration such that the sections  $s[j]$  to  $e[j]$  are covered and nothing else is covered. More formally, for each  $1 \leq x \leq n$ ,

- If  $s[j] \leq x \leq e[j]$ , section  $x$  is covered.
- Otherwise, section  $x$  is not covered.

For each of these  $q$  performances, help Benson to determine if there exists a curtain configuration satisfying his requirements.

### Input format

Your program must read from standard input.

The first line of input will contain 3 spaced integers  $n$ ,  $m$  and  $q$ , representing the number of sections, curtains and performances respectively.

The next  $m$  lines of input will contain 2 spaced integers each. The  $i$ -th of these lines will contain  $l[i]$  and  $r[i]$  respectively, describing the range of sections that curtain  $i$  can cover.

The next  $q$  lines of input will contain 2 spaced integers each. The  $j$ -th of these lines will contain  $s[j]$  and  $e[j]$  respectively, describing the range of sections that need to be covered for performance  $j$ .

### Output format

Output  $q$  lines, the  $j$ -th of which should contain YES if it is possible to cover the required sections for the  $j$ -th performance using the curtains, and NO otherwise.



## Subtasks

For all subtasks, it is guaranteed that:

- $1 \leq n, m, q \leq 500\,000$
- $1 \leq l[i] \leq r[i] \leq n$  (for all  $1 \leq i \leq m$ )
- $1 \leq s[j] \leq e[j] \leq n$  (for all  $1 \leq j \leq q$ )

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Marks	Additional Constraints
1	3	$1 \leq n, m, q \leq 200$
2	6	$1 \leq n, m, q \leq 2000$
3	15	$1 \leq n \leq 2000$
4	20	$s[j] = 1$
5	36	$1 \leq n, m, q \leq 100\,000$
6	20	No additional restrictions

## Sample Testcase 1

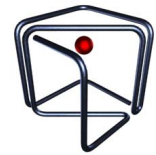
This testcase is valid for all subtasks.

Input	Output
6 2 3	NO
1 2	YES
3 4	NO
1 3	
1 4	
1 5	

## Sample Testcase 1 Explanation

Benson has 6 sections and 2 curtains. Curtain 1 covers sections 1 and 2 while curtain 2 covers sections 3 and 4.

It is not possible to exactly cover sections 1 to 3. It is also not possible to exactly cover sections 1 to 5. However, he can use both curtains to cover sections 1 to 4 exactly.



## Sample Testcase 2

Input	Output
10 10 10	NO
6 9	NO
6 7	YES
1 6	NO
10 10	YES
5 9	NO
3 9	NO
2 10	NO
5 7	NO
9 10	YES
5 10	
7 8	
4 7	
1 6	
2 7	
3 9	
7 7	
2 9	
4 9	
6 6	
5 7	