

Kick Start 2019 - Round A

Contention

Problem

You are selling tickets for the front row of seats at a movie theater. The front row has N seats, numbered 1 to N from left to right. You have been out of the office the last week, and upon your return, Q bookings for seats have piled up! The i -th booking requests all the seats from L_i to R_i inclusive. You now have the boring job of entering each booking into the system, one at a time.

Since some of the bookings may overlap, the system might not be able to fulfill each booking entirely. When you enter a booking into the system, it will assign every seat requested by the booking that hasn't already been assigned to a booking entered into the system earlier.

What is the largest integer k where there exists an order that you can enter the bookings into the system, such that each booking is assigned at least k seats?

Input

The first line of the input gives the number of test cases, T . T test cases follow. Each test case starts with a line containing two integers N and Q , the number of seats and the number of bookings, respectively. Then, there are Q more lines, the i -th of which contains the two integers L_i and R_i , indicating that the i -th booking would like to book all the seats from L_i to R_i , inclusive.

Output

For each test case, output one line containing `Case #x: y`, where x is the test case number (starting from 1) and y is the largest value k , as described above.

Limits

Time limit: 15 seconds per test set.

Memory limit: 1GB.

$T = 100$.

$1 \leq N \leq 10^6$.

$1 \leq L_i \leq R_i \leq N$.

Test set 1 (Visible)

$1 \leq Q \leq 300$.

Test set 2 (Hidden)

$1 \leq Q \leq 30000$.

For at least 85 of the test cases, $Q \leq 3000$.

Sample

Sample Input

Sample Output

```
3
5 3
1 2
3 4
2 5
30 3
10 11
10 10
11 11
10 4
1 8
4 5
3 6
2 7
```

```
Case #1: 1
Case #2: 0
Case #3: 2
```

In Sample Case #1, there are **N** = 5 seats and **Q** = 3 bookings. One possible order is:

- Put in the second booking, where the system will book 2 seats (3 and 4).
- Put in the first booking, where the system will book 2 seats (1 and 2).
- Put in the third booking, where the system will book 1 seat (only seat 5, since seats 1, 2, 3 and 4 are already booked).

Each booking is assigned at least 1 seat, and there is no order that assigns at least 2 seats to each booking, so the answer is 1.

In Sample Case #2, there are **N** = 30 seats and **Q** = 3 bookings. No matter what order you assign the seats in, at least one booking will have no seats assigned to it. So the answer is 0. Notice that there can be seats that are not part of any bookings!

In Sample Case #3, there are **N** = 10 seats and **Q** = 4 bookings. One possible order is:

- Put in the second booking, where the system will book 2 seats (4 and 5).
- Put in the third booking, where the system will book 2 seats (3 and 6, since 4 and 5 are already booked). Notice that the seats booked are not necessarily adjacent to each other.
- Put in the fourth booking, where the system will book 2 seats (2 and 7).
- Put in the first booking, where the system will book 2 seats (1 and 8).

Each booking is assigned at least 2 seats, and there is no order that assigns at least 3 seats to each booking, so the answer is 2.

Note: We do not recommend using interpreted/slower languages for the Large dataset of this problem.