Code Jam 2017 - Round 2

Roller Coaster Scheduling

Problem

You created a new roller coaster that is about to open. Its train consists of a single row of **N** seats numbered 1 through **N** from front to back. Of course, seats closer to the front are more valuable. Customers have already purchased opening-day tickets. Each ticket allows a specific customer to take one ride on the coaster in a particular seat. Some customers may have bought more than one ticket, and they expect to go on one ride for each ticket.

You need to decide how many roller coaster rides there will be on opening day. On each ride, one customer can sit in each seat; some seats on a ride might be left empty. You cannot assign a customer to more than one seat in the same ride, nor can you put two customers on the same seat in any given ride.

You wish to minimize the number of rides required to honor all tickets, to reduce operational costs. To reduce the required number of rides, you can *promote* any number of tickets. Promoting a ticket means taking a customer's ticket and giving that customer a new ticket for a seat closer to the front of the train (that is, a seat with a lower number). You would prefer to promote as few tickets as possible, since too many promotions might cause customers to get greedy and ask for more promotions in the future.

Given the positions and buyers of all the tickets that have been sold, what is the minimum number of rides needed to honor all tickets, using as many promotions as needed and scheduling the rides optimally? And what is the minimum number of ticket promotions necessary to attain that number of rides? Note that promoting a given customer on a given ride from seat 4 to seat 2, for example, counts as only one promotion, not two separate ones.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case starts with a single line with three integers **N**, the number of seats in the roller coaster, **C**, the number of potential customers, and **M**, the number of tickets sold. The customers are identified with numbers between 1 and **C**. Then, **M** lines follow, each containing two integers: $\mathbf{P}_\mathbf{i}$, the position in the roller coaster assigned to the i-th ticket, and $\mathbf{B}_\mathbf{i}$, the identifier of the buyer of that ticket.

Output

For each test case, output one line containing `Case #x: y z`, where `x` is the test case number (starting from 1), `y` is the minimum number of rides you need to honor all tickets if you use the promotions and schedule the rides optimally, and `z` is the minimum number of promotions you need to make be able to honor all tickets with `y` rides.

Limits

Time limit: 20 seconds per test set.
Memory limit: 1 GB.
1 ≤ **T** ≤ 100.
2 ≤ **N** ≤ 1000.
1 ≤ **M** ≤ 1000.

1 ≤ $\mathbf{P_i}$ ≤ **N**.
1 ≤ $\mathbf{B_i}$ ≤ **C**.

Small dataset (Test Set 1 - Visible)

C = 2.

Large dataset (Test Set 2 - Hidden)

2 ≤ **C** ≤ 1000.

Sample

Sample Input

```
5
2 2 2
2 1
2 2
2 2 2
1 1
1 2
2 2 2
1 1
2 1
1000 1000 4
3 2
2 1
3 3
3 1
3 3 5
3 1
2 2
3 3
2 2
3 1
```

Sample Output

```
Case #1: 1 1
Case #2: 2 0
Case #3: 2 0
Case #4: 2 1
Case #5: 2 1
```

Note that the last two sample cases would not appear in the Small dataset.

In Case #1, both customers purchased a ticket for position 2. It is impossible to honor both tickets with a single ride, but promoting either ticket to position 1 allows you to accommodate both tickets on the same round.

Case #2 is a similar story, except both tickets are for position 1. Since you cannot promote those tickets or exchange them for inferior tickets, you are forced to run 2 separate rides, one per customer.

Case #3 features the same customer purchasing both positions. Since you are forced to have 2 rides for that customer, there is no reason to give out any promotions.

In Case #4, notice that there may be both customers and positions with no tickets assigned. In this case, there are three tickets sold for position three. If you promote customer 2 to position 2, for instance, you can have one ride with customer 1 sitting in position 2 and customer 3 sitting in position 3, and a second ride with customer 2 in position 2 and customer 1 in position 3.

Additional promotions will not allow you to decrease the number of rides, because customer 1 has two tickets and you need to honor those in different rides, regardless of position.

In Case #5, one optimal solution is to promote one of the `3 1` tickets to `1 1`.