Exchanges

Given n integer registers r_1, r_2, ... , r_n we define a Compare-Exchange Instruction CE(a,b), where a, b are register indices (1 <= a < b <= n):

```
CE(a, b)::
  if content(r_a) >  content(r_b) then
     exchange the contents of registers r_a and r_b;
```

A Compare-Exchange program (shortly CE-program) is any finite sequence of Compare-Exchange instructions. A CE-program is called a Minimum-Finding program if after its execution the register r_1 always contains the smallest value among all values in the registers. Such a program is called reliable if it remains a Minimum-Finding program after removing any single Compare-Exchange instruction. Given a CE-program P, what is the smallest number of instructions that should be added at the end of program P in order to get a reliable Minimum-Finding program?

For instance, consider the following CE-program for 3 registers: CE(1, 2), CE(2, 3), CE(1, 2). In order to make this program a reliable Minimum-Finding program it is sufficient to add only two instructions: CE(1, 3) and CE(1, 2).

Task

Write a program that:

- reads the description of a CE-program,
- computes the smallest number of CE-instructions that should be added to make this program a reliable Minimum-Finding program,
- writes the result.

Input

The first line of the input contains exactly one positive integer d equal to the number of data sets, 1 <= d <= 10. The data sets follow.

Each data set consists of exactly two consecutive lines. The first of those lines contains exactly two integers n and m separated by a single space, 2 <= n <= 10000, 0 <= m <= 25000. Integer n is the number of registers and integer m is the number of program instructions.

The second of those lines contains exactly 2m integers separated by single spaces - the program itself. Integers a_j, b_j on positions 2j-1 and 2j, 1 <= j < = m, 1 < = a_j < b_j <= n, are parameters of the j-th instruction in the program.

Output

The output should consist of exactly d lines, one line for each data set. Line i, 1 <= i <= d, should contain only one integer - the smallest number of instructions that should be added at the end of the i-th input program in order to make this program a reliable Minimum-Finding program.

Example

Sample input:
1
3 3
1 2 2 3 1 2

Sample output:
2