UVa Online Judge

1117 Reliable Programs

Consider a machine with n integer registers $r_1, r_2, \ldots, r_n$ and a single type of *compare-exchange instruction*, CE(i, j) defined as follows, where $1 \leq i < j \leq n$ are the register indices:

CE(i, j): if $content(r_i) > content(r_j)$
then exchange the contents of registers r_i and r_j.

A compare-exchange program (shortly CE-program) is any finite sequence of compare-exchange instructions. A CE-program is called *minimum-finding* if after its execution the register r_1 always contains the minimum value among all the initial values in the registers. Furthermore, such 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 minimum number of instructions that should be added at the end of program P in order to make it reliable?

Example

Consider the following 3-register CE-program: CE(1,2); CE(2,3); CE(1,2).

In order to make this program reliable it suffices to add only two extra instructions, namely CE(1,3) and CE(1,2).

Write a program that reads the description of a CE-program, and determines the minimum number of CE-instructions that should be added to make this program reliable.

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

The first line of input contains the number of registers n, where $0 \leq n \leq 1000$, followed by the number of program instructions m, where $0 \leq m \leq 3000$.

The next line contains the program itself: a sequence of $2m$ integers, separated by spaces, where each CE-instruction consists of two consecutive integers on positions $2j - 1$ and $2j$, with $1 \leq j \leq m$.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

The output consists of a single integer: the minimal number of instructions that should be added to the input program in order to make this program reliable.

Sample Input

```
1

3 3
1 2 2 3 1 2
```

Sample Output

```
2
```