

## 5062 Transitive Closure

In almost all ICPC trainings, one of the basic items to be taught (or even better, assumed) is how to compute the transitive closure of a directed graph. Since this concept is so elementary, there is a professor that suggests to solve this problem using Warshall's algorithm. As one of the bonus tasks of this ICPC, we would like to examine whether the participants have indeed mastered this technique.

Given a directed graph  $G = \langle V, E \rangle$  with  $N = |V|$  vertices ( $2 \leq N \leq 2,500$ ) and  $M = |E|$  edges ( $1 \leq M \leq 10,000$ ), its transitive closure is the binary relation  $R$  on  $V$  such that two vertices,  $A$  and  $B$ , are related in  $R$  if there is a directed path from  $A$  to  $B$ , where a directed path is a sequence of edges of the graph  $C_0C_1, C_1C_2, \dots, C_{K-1}C_K$  such that  $A = C_0$  and  $B = C_K$ . Your task is to compute  $R$ .

Because  $R$  might be very large, we've decided to simplify the output. Calculate the number of ordered pairs of  $(X, Y)$  where  $X$  is not equal to  $Y$  such that there is a path from vertex  $X$  to vertex  $Y$  in graph  $G$ .

### Input

The first line of input contains an integer  $T$  ( $T \leq 10$ ), denoting the number of testcases. The following lines describe each test case of the following format.

The first line of a testcase consists of two integers  $N$  and  $M$ . Each of the following  $M$  lines contains 2 integers,  $A$  and  $B$ , denoting that there is a directed edge from  $A$  to  $B$ . The vertices are numbered from 1 to  $N$ , and there will be no repeated edges within a description of a graph.

### Output

For each case, output an integer denoting the number of ordered pairs of  $(X, Y)$  where  $X$  is not equal to  $Y$  such that there is a path from vertex  $X$  to vertex  $Y$ .

**Note:** Explanation for the 1-st sample input.

The binary relation  $R$  is:

```
0 1 1
0 0 1
0 0 0
```

There are 3 ordered pairs of  $(X, Y)$  such that  $R_{X,Y}$  is true.

### Sample Input

```
2
3 2
1 2
2 3
4 4
1 2
2 3
2 4
4 1
```

**Sample Output**

3  
9