

# BOI 97 - Task Execution

Assume that we have a number of tasks that must be executed. However, the tasks are not independent to each other. We say that task 2 depends on task 1, if the execution of task2 can start after the completion of task 1. However, we may find tasks that at any given time can be executed in parallel, saving time. Given a number of tasks and their dependencies, determine the shortest time that all tasks can be executed in a computer with an infinite number of processors. Then, you are requested to determine the minimum number of processors in order to execute the tasks in the (previously found) shortest time. Each task takes 1 time unit of execution. The tasks are represented by positive integers from 1 to N ( $N \leq 200$ ).

## Input

The first line contains a positive integer number N denoting the number of tasks to be executed in the computer, and another positive integer number M, denoting the number of dependencies. The next M lines until the end of the input file, contain the dependencies between the tasks. For example, when the input line, which corresponds to a dependency, contains the string "2 3", this means that in order to start the execution of task 3, task 2 must be completed first. Data is always correct and there is always a solution.

## Output

Output will consist only one line, which contains two positive integer numbers, separated by space character. The first number T denotes the minimum number of time units that we need in order to execute all tasks, assuming an infinite number of processors. The second number denotes the minimum number of processors we can use in order to execute the tasks in T time units.

## Example

**Input:**

```
6 6
1 4
2 5
3 6
4 6
4 5
5 6
```

**Output:**

```
4 2
```