ETH Zurich
Institut für Theoretische Informatik
Prof. Dr. Angelika Steger

HS 2014
Exercise Sheet 7

Algorithms Lab

Exercise 1 – *Dress the professor*

Description Professor Smith wears the same (but clean!) combination of clothes everyday. His combination of clothing consists of n pieces denoted by $c_1, \ldots, c_n$. However, he hates when in the morning he puts his hat first only to remember that he still didn't put his sweater on. We say that a piece of clothing c_i depends on a piece of clothing c_j if the professor must put on c_j before putting on c_i. Being an analytical person, he realised that each piece of clothing can depend on some other pieces (even more than one) of clothing, but that it is possible for him to dress up completely. In other words, professor can put on all $c_1, \ldots, c_n$ and he will never put on c_i before c_j if c_i depends on c_j. Your task is to help the professor and determine the order in which he should put on his clothes. As there could be more than one possible solution, you should output *lexicographically biggest*.

Input The first line of input contains T ($1 \leq T \leq 5$), the number of test-cases to process. Each test case starts with a line consisting of two natural number n and m ($1 \leq m \leq 50n$), the number of pieces of clothing and the number of dependencies, respectively.

Each of the next m lines of the test case consists of a pair of integers i and j , $1 \leq i, j \leq n$, which means that c_i depends on c_j.

Remark: For the "small" test set, you can assume that $n \leq 1000$, and for the "large" test set $n \leq 70000$.

Output For each test case you have to output n integers on separate lines. Line $i + 1$ of the test case should contain index of i-th piece of clothing the professor should put on. Therefore, if you output integers $i_1, \ldots, i_n$, that means that the professor should put on clothes in the order $c_{i_1}, c_{i_2}, c_{i_3}, \ldots, c_{i_n}$.

Sample input

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

Sample output

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