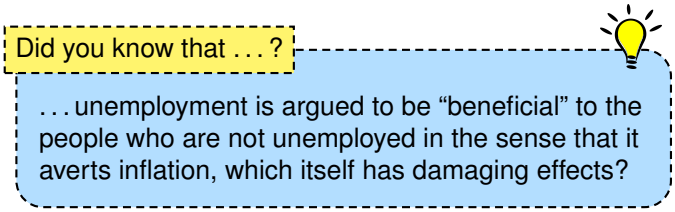
Problem ID: jugglingjobs

Last week, Rudi was almost fired from his job. Before his boss left for an important meeting, he handed Rudi a hand-written list of tasks that he wanted him to take care of while he was attending the meeting. Unfortunately, his boss wrote down the tasks in no particular order, and Rudi did not realise that some of these tasks had to be completed before he could start working on another task: for example, before he can send an important e-mail to a colleague, he first needs to figure out the colleague's e-mail address. He also needs to formulate the content of the e-mail, but it does not matter whether he does that before or after looking up the colleague's e-mail address.



After he returned from his meeting, Rudi's boss realised that due to the lack of clear instructions, Rudi had not managed to complete all tasks. Disappointed, he started writing down a list of task dependencies, detailing pairs of tasks where the first task needs to be finished before the second one can be started. He also told Rudi that if he was not able to finish all tasks on the current work day, he would be fired.

Now, Rudi has a problem. There are only 5 hours left in the current work day, and he still has a lot of work to do! The amount of work is so large that there is no way that Rudi could complete it all on his own in the remaining time. Luckily, Rudi is a very popular person and has an infinite number of friends that he can count on. After he called all of his friends, and all of them gladly agreed to help him out, he now needs your help to figure out which tasks his friends can execute simultaneously.

Since Rudi has an unlimited number of friends, they can execute as many tasks at the same time as needed, but only if all of each task's prerequisites have been fulfilled. Additionally, as Rudi would like to leave the office and go home as early as possible, he wants his friends to start working on his tasks as soon as possible. Since the tasks assigned to Rudi are very simple, any of Rudi's friends can perform any task, and all tasks take the same time to finish.

Input

The input consists of:

- One line with two integers n and m ($1 \leq n \leq 10^5$, $0 \leq m \leq 2 \cdot 10^5$), where n is the number of tasks and m is the number of constraints on the order of the tasks. The tasks are numbered from 1 to n.
- m lines, each with two integers a and b ($1 \leq a, b \leq n$), indicating that task a must be completed before task b can be executed.

It is guaranteed that there exists some order in which all n tasks can be completed.

Output

Assume that Rudi's friends carry out the tasks in an optimal way. For each time step, output a line with the numbers of all tasks that are performed in this time step. The tasks on each row can be output in any order.

Sample Input 1

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

Sample Output 1

```
1 5
2 3 4
```

Sample Input 2

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

Sample Output 2

```
1 2
3
```