

## 11469 Treasure Hunt

Famous Archaeologist Dr. Asiana Jones has discovered something extraordinary. He found a thousand years old tomb, and discovered that there are thousands of treasure rooms beneath it. Each treasure room contains a chest full of treasure. There can be a number of tunnels to go from one room to another. But, since the tunnels are thousand years old (like the tomb itself), they are very fragile. You have to be very quick, and you can't use a tunnel more than once.

Dr. Asiana Jones has managed to calculate the exact positions of all the tombs. But, as the rooms are beneath the tomb, he must dig a hole, one the floor to get to any of them. He doesn't want to dig more than one hole, because, otherwise, the whole tomb may become unstable.

If Dr. Jones can start from any of the treasure room, find the maximum number of treasure he can collect. Dr. Jones has to start and end at the same room, since, that's the only way to get out of the tomb.

### Input

First line of input contains an integer  $T$ , the number of test cases, followed by  $T$  scenarios. Each scenario starts with two integers,  $N$  and  $M$ , the number of treasure rooms, and the number of tunnels.

Each treasure room is uniquely identified by an integer between 1 and  $N$ . Each of the following  $M$  lines each, contain two integers, the treasure rooms at the end of the corresponding tunnel. The tunnels can be traversed in any direction, but at most once. That is, if there is a tunnel between room 1 and 2, it can be used to move from room 1 to 2, or 2 to 1. But, after any of these move, you can't use this tunnel any more. There will be a blank line before each scenario.

### Output

For each scenario, print the scenario number, followed by the maximum number of treasure to collect. Next line will contain a sequence of integers, the starting rooms, from where, you can collect maximum number of treasures.

### Constraints:

- $T \leq 31$
- $1 \leq N \leq 10,000$
- $0 \leq M \leq 1,000,000$

### Sample Input

```
3

4 3
1 2
2 3
1 4

3 3
1 2
2 3
```

3 1

4 4

1 2

2 3

3 4

4 2

### Sample Output

Case #1: 1

1 2 3 4

Case #2: 3

1 2 3

Case #3: 3

2 3 4