

Distinct Viewpoints

The citizens of Palinopolis are very fond of arguments. Whenever two Palinopolitans meet up, whether it be to negotiate over territorial boundaries or buy a pet, custom requires them to take up contrary positions and argue for at least five minutes, then finally depart in frustrated disagreement. The most gifted arguers often pursue careers at prestigious argument firms, but in order to secure a job they must first survive a stringent regime of exams followed by a highly competitive hiring process. In the first round of hiring, each applicant must face off against the head of the firm for ten hours without agreeing on a single point. In the second round, the remaining applicants face each other. These applicants are conveniently numbered from 1 to N , and a topic is chosen that allows N mutually opposing viewpoints V_1, V_2, \dots, V_N to be assigned to them respectively. Next, pairs of applicants are successively chosen and locked in a small, humid room to argue their respective viewpoints until one of them becomes conciliatory, gives up, or passes out. The loser is not eliminated outright but must take on the viewpoint of the victor in subsequent matches. Additionally, anyone else who is currently assigned the losing opponent's viewpoint must change his viewpoint to the victor's. It may happen that some pairs are chosen such that each opponent represents the same viewpoint. This is in fact no problem at all, since any Palinopolitan worth his salt is perfectly capable of conducting a heated argument with someone who is expressing the exact same views. No applicant is ever asked to argue against himself, however, as that would just be silly.

The final selection of applicants is based on many complex factors, including percentage of matches won, number of hours passed out, and whether or not the judge has had a good lunch that day. As a member of the oversight committee, part of your job is to analyse the matches and provide relevant figures and statistics to your superiors. One crucial bit of information (according to your boss) is how many viewpoints survived the process, and which groups of applicants share the same assigned viewpoint. This would normally be quite an easy task for you to solve, except that today your boss is in a bad mood and would like to punish you by making you solve it in a strange, primitive programming language that he just read about on the internet when he was supposed to be working.

Note: You can use any programming language you want, as long as it is brainf**k.

Input

First an integer T ($1 \leq T \leq 1000$), followed by T tests. Each test starts with two integers N ($1 \leq N \leq 50$) and M on a line, the number of applicants and the number of matches respectively. Then in the following M lines are given pairs of integers u and v in the range $[1..N]$, indicating that applicants u and v face each other in a match. Finally, every case (including the last) ends with a blank line.

Output

For each case, first print out K , the number of distinct viewpoints surviving at the end of all matches, and then print out each group of applicants sharing a common viewpoint. Each group must be sorted internally, and the groups must also be sorted with respect to each other, using the usual definitions for sorting lists (lexicographically).

Example

Input:

```
2
3 1
2 3
```

```
9 7
4 1
8 2
4 6
5 1
5 7
6 1
5 6
```

Output:

```
2
1
2 3
4
1 4 5 6 7
2 8
3
9
```

Additional Info

The input is generated as follows: **N** is chosen uniform randomly in the range, then **K** is chosen uniform randomly in $[1..N]$ and a minimal number of edges is chosen randomly to yield **K**, then the extra edges (that do not change **K**) are chosen randomly with probability **p**, where **p** is uniform random in $[0,1]$. Duplicate edges do not appear.

There are two data sets, one with **T**=1000 and the other with **T**=500. My solution at the time of publication has 1192 bytes (not golfed) and runs in 9.37s with 2.3M memory footprint.

The custom judge reports your source code length in parentheses after the judge result (when available at the time of judge termination). Only valid BF commands are counted. Also, if you don't get AC, you will be told on which data set you first failed (0-indexed). (The master judge is the same as [BFPRMCYC](#), and the test case judge is the same as [BFBINADD](#).)