Taking Over The World

Problem

You and your friend Pinky have a plan to take over the world. But first you need to disable a certain secret weapon.

It is hidden inside a twisted maze of passages (a graph) with one entrance. Pinky is going to be at the vertex with the secret weapon, disabling it. Meanwhile, a security team at the graph entrance will be alerted, and will run through the graph to try to get to Pinky in time to stop him. You are going to be slowing down the security team to give Pinky as much time as possible. It takes one unit of time to traverse any edge of the graph, but you can additionally "obstruct" up to **K** vertices. It takes one additional unit of time to traverse an obstructed vertex. You will choose to obstruct a set of vertices that slows down the security team by as much as possible.

If the security team will be starting at the graph entrance and is trying to get to the secret weapon vertex, how much time will it take them to get there? Note that you have to commit all your obstructions before the security guards start their journey, and the security guards will know which vertices you have obstructed and will choose an optimal path based on that information.

Obstructing the secret weapon vertex is not useful because that will not delay the guards any further after they have already caught Pinky. Obstructing the entrance, on the other hand, is obviously a good idea.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each one starts with a line containing **N**, **M**, and **K**. The next **M** lines each contain a pair of vertices connected by an edge. Vertices are numbered from 0 (the entrance) to **N** - 1 (the secret weapon room). The first vertex will always be smaller than the second vertex, and no pair of vertices will appear more than once in the same test case. Edges are bi-directional -- the guards can travel along any edge in either direction.

Output

For each test case, output one line containing "Case #x: y", where x is the test case number (starting from 1) and y is the time it will take the security guards to get from the entrance to the secret weapon room.

Limits

Memory limit: 1 GB.
1 ≤ **T** ≤ 100.
2 ≤ **N** ≤ 100.
1 ≤ **M** ≤ **N** * (**N** - 1) / 2.
1 ≤ **K** ≤ **N**.
There will always be a path from room 0 to room **N** - 1.

Small dataset

Time limit: 240 seconds.
It will not be possible to delay the guards by more than 2 time units, compared to the shortest
unobstructed path length (using the given **K**).

Large dataset

Time limit: 480 seconds.
No extra restrictions.

Sample

<table>
<tr><td>

Sample Input

```
5
3 2 1
0 1
1 2
3 2 2
0 1
1 2
3 2 3
0 1
1 2
4 4 2
0 1
0 2
1 3
2 3
7 11 3
0 1
0 2
0 3
1 4
1 5
2 4
2 5
3 4
3 5
4 6
5 6
```

</td><td>

Sample Output

```
Case #1: 3
Case #2: 4
Case #3: 4
Case #4: 3
Case #5: 5
```

</td></tr>
</table>