1291 – Real Life Traffic

Dhaka city is full of traffic jam and when it rains, some of the roads become unusable. So, you are asked to redesign the traffic system of the city such that if exactly one of the roads becomes unusable, it's still possible to move from any place to another using other roads.

You can assume that Dhaka is a city containing some places and bi directional roads connecting the places and it's possible to go from any place to another using the roads. There can be at most one road between two places. And of course there is no road that connects a place to itself. To be more specific there are **n** places in Dhaka city and for simplicity, assume that they are numbered from **0** to **n-1** and there are **m** roads connecting the places.

Your plan is to build some new roads, but you don't want to build a road between two places where a road already exists. You want to build the roads such that if any road becomes unusable, there should be an alternate way to go from any place to another using other roads except that damaged road. As you are a programmer, you want to find the minimum number of roads that you have to build to make the traffic system as stated above.

Input

Input starts with an integer **T (≤ 30)**, denoting the number of test cases.

Each case starts with a blank line. The next line contains two integers: **n (3 ≤ n ≤ 10000)** and **m (≤ 20000)**. Each of the next **m** lines contains two integers **u v (0 ≤ u, v < n, u ≠ v)** meaning that there is a bidirectional road between place **u** and **v**. The input follows the above constraints.

Output

For each case, print the case number and the minimum number of roads you have to build such that if one road goes down, it's still possible to go from any place to another.

Sample Input	Output for Sample Input
2 4 3 1 2 2 3 2 0 3 3 1 2 2 0 0 1	Case 1: 2 Case 2: 0

Note

1. Dataset is huge, use faster I/O methods.
2. For case 1, one of the solutions is to construct two roads in (0, 1) and (1, 3).