Don't Really Like How The Story Ends

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There are n planets in the galaxy, and many undirected warp tunnels connecting them. 6000 years ago, Spinel performed a depth-first search on the planets, visited all of them, and labeled them from 1 to n in the order of discovery.

Many warp tunnels have broken down since, and only m of them are still working. Spinel wants to know how many new warp tunnels have to be built so that it is possible to perform a depth-first search, where the order of discovery is exactly as labeled 6000 years ago.

Recall that the depth-first search (DFS) algorithm inputs a graph G and a vertex v of G, and labels all vertices reachable from v as discovered.

Here is the pseudocode of a recursive implementation of DFS:

```
procedure DFS(G, v) is
    label v as discovered
    for all vertices w that there exists an edge between v and w do
        if vertex w is not labeled as discovered then
            recursively call DFS(G, w)
```

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \leq n, m \leq 10^5$) indicating the number of planets and the number of remaining warp tunnels.

For the following m lines, the i-th line contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n$) indicating a warp tunnel between u_i and v_i.

It's guaranteed that the sum of $(n + m)$ of all test cases will not exceed 10^6.

Output

For each test case output one line containing one integer, indicating the minimum number of new warp tunnels that have to be built.

Example

standard input	standard output
3 2 3 1 1 1 2 2 1 4 1 1 4 4 2 1 2 3 4	0 2 1

Note

For the second sample test case we can add a tunnel between planet 1 and 2, and add another tunnel between planet 2 and 3.

For the third sample test case we can add a tunnel between planet 2 and 3.