3 Checking Whether Any Intersection in a City is Reachable from Any Other

Problem Introduction

The police department of a city has made all streets one-way. You would like to check whether it is still possible to drive legally from any intersection to any other intersection. For this, you construct a directed graph: vertices are intersections, there is an edge (u, v) whenever there is a (one-way) street from u to v in the city. Then, it suffices to check whether all the vertices in the graph lie in the same strongly connected component.

Problem Description

Task. Compute the number of strongly connected components of a given directed graph with n vertices and m edges.

Input Format. A graph is given in the standard format.

Constraints. $1 \leq n \leq 10^4$, $0 \leq m \leq 10^4$.

Output Format. Output the number of strongly connected components.

Time Limits.

language	C	C++	Java	Python	C#	Haskell	JavaScript	Ruby	Scala
time (sec)	1	1	1.5	5	1.5	2	5	5	3

Memory Limit. 512MB.

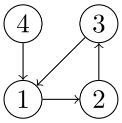
Sample 1.

Input:

```
4 4
1 2
4 1
2 3
3 1
```

Output:

```
2
```



This graph has two strongly connected components: $\{1, 3, 2\}$, $\{4\}$.

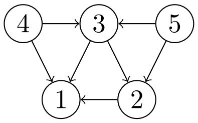
Sample 2.

Input:

```
5 7
2 1
3 2
3 1
4 3
4 1
5 2
5 3
```

Output:

```
5
```



This graph has five strongly connected components: {1}, {2}, {3}, {4}, {5}.