Ada and Panels

Ada the Ladybug has proved succesful in solving some hard problems, so a construction company has asked her to solve a problem for them. There are multiple cities in the country and each city owns exactly one panel-block. There are also unidirectional roads between some pairs of cities (note that circular self-roads and multi-roads with several traffic lines are allowed). A city can sell its panel-block to any city, to which they could transport the panel block, and from which they can bring back their reward for it (i.e. there must be a path from actual city to destination city and back). As long a city has **K** panel-blocks, it builds a prefab of height **K** which looks exactly same as each other prefab of height **K**.

The construction company notes all the heights down and puts them in an array. They are wondering how many distinct arrays are possible by moving the panel blocks between cities. However there is a catch. Consider a set of cities in which each city is reachable by every other city. Since we can easily shuffle the panel blocks between such a set of cities, we can create new permutations with the same set of heights. The construction company will NOT count any such cases. Hence they will only consider the distinct set of heights for such a set of cities.

You have proved succesful in helping Ada with some hard problems, so she has asked you to help her. Your job is following - count the number of possible structures which could arise. Since this number might be pretty big, you have to output it modulo $\mathbf{10^9+7}$.

Input

The first line contains two integers $\mathbf{1 \le N, M \le 2*10^5}$, the number of cities an the number of unidireectional roads between them.

The next **M** lines contains two integers $\mathbf{0 \le a, b < N}$, the road from city **a** to city **b**

Output

Print a single line - the number of possible structures modulo **1000000007**.

Example Input

```
7 9
0 1
1 2
2 3
3 1
2 0
4 5
5 4
6 4
5 6
```

Example Output

```
15
```

Example Input 2

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

Example Output 2

```
15
```

Example Input 3

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

Example Output 3

```
1
```

Example Input 4

```
8 7
0 1
1 0
2 3
3 2
4 5
5 6
6 4
```

Example Output 4

```
12
```