I: Emails



Time limit: 3 seconds



Ariadna's blog is filled with delicious recipes and sensible advice for a healthy and balanced lifestyle. Unsurprisingly, it has thus gathered an impressive number of readers. This reader base is now stable, and Ariadna feels that it would be useful for them to interact more and form a tighter community, one that is not solely anchored to the blog.

Ariadna knows that some of the readers are already friends or acquaintances, and therefore have each other's email addresses. She thinks that a good start for developing the community would be for everyone to have everyone else's email address, so that everyone would be able to reach out to the entire group. Since she knows her blog's readers also greatly enjoy doing things in a "decentralized" fashion, she therefore devises the following protocol, to be started on day D:

- Every day at 8am, everyone sends the current list of contacts in their address book to all of the contacts in their address book.
- Every day at 8pm, everyone updates their address book, adding any new received email addresses.

If a person does not need to do any update at 8pm, then the process is said to have *converged* for this person, and she will no longer need to continue sending emails over the next days.

You are a skillful hacker and you have managed to get access to all of the blog readers' address books. You would like to surprise and impress Ariadna by notifying her of whether or not the process she proposes will lead to everyone getting everyone else's address. Moreover, if the process is meant to succeed, you want to give her a good estimate of how many days it would take. More precisely, if the process succeeds, you can either give her:

- the number E of days (including the first day) elapsed until the last update takes place, or
- the number of days (including the first day) elapsed until the process has *converged* on everyone's side. Note that, according to Ariadna's definition, this is equal to E+1.

Input

The first line of the input contains two integers N and M, corresponding to the number of readers and respectively to the number of pairs of readers that initially have each other's email address. Readers are numbered from 1 to N.

The M following lines each contain two integers, i and j, meaning that readers i and j initially have each other's email address. Note that this means that both reader i has reader j's address and reader j has reader i's address.

Output

The output should contain a single integer equal to either:

- -1 if the process does not lead to everyone eventually having everyone else's email address, or
- the estimated necessary number of days, otherwise. Note that this number may be equal to 0.

Notes

- We assume the reader base is stable, i.e. no reader leaves and no additional reader joins throughout the process.
- We assume that everyone knows their own email address; receiving one's own address is simply ignored.
- You do no have to be "consistent" in your answers across several tests cases, meaning that you can output the value E for one test case and E+1 for another.

Limits

- $2 \leqslant N \leqslant 100\,000$
- $1 \leqslant M \leqslant 100\,000$

Sample Input 1

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

Sample Output 1

```
2
```

Sample Explanation 1

The process proceeds as follows:

- On day D at 8am:
  - Reader 1 sends the address of reader 2 to reader 2.
  - Reader 2 sends the addresses of readers 1 and 3 to readers 1 and 3.
  - Reader 3 sends the addresses of readers 2 and 4 to readers 2 and 4.
  - Reader 4 sends the address of reader 3 to reader 3.
- On day D after the 8pm update:
  - Reader 1's address-book has been updated and contains the addresses of readers 2 and 3.
  - Reader 2's address-book has been updated and contains the addresses of readers 1, 3 and 4.
  - Reader 3's address-book has been updated and contains the addresses of readers 1, 2 and 4.
  - Reader 4's address-book has been updated and contains the addresses of readers 2 and 3.
- On day D+1 at 8am:
  - Reader 1 sends the addresses of readers 2 and 3 to readers 2 and 3.
  - Reader 2 sends the addresses of readers 1, 3 and 4 to readers 1, 3 and 4.
  - Reader 3 sends the addresses of readers 1, 2 and 4 to readers 1, 2 and 4.
  - Reader 4 sends the addresses of readers 2 and 3 to readers 2 and 3.
- On day D+1 after the 8pm update:

  - Reader 1's address-book has been updated and contains the addresses of readers 2, 3 and 4.
  - The process has converged for reader 2 since there is no update.
  - The process has converged for reader 3 since there is no update.
  - Reader 4's address-book has been updated and contains the addresses of readers 1, 2 and 3.
- On day D+2 at 8am:
  - Reader 1 sends the addresses of readers 2, 3 and 4 to readers 2, 3 and 4.
  - Reader 4 sends the addresses of readers 1, 2 and 3 to readers 1, 2 and 3.
- On day D+2 after the 8pm update:
  - The process has converged for reader 1 since there is no update.
  - The process has converged for reader 4 since there is no update.

The last update takes place on day D+1, after **2** elapsed days. The process has converged for everyone on day D+2, after **3** elapsed days. The sample output contains the former value, **2**. Outputting the latter value, **3**, is an equally correct alternative.

Sample Input 2

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

Sample Output 2

```
-1
```

Page intentionally left blank