Czech ACM Student Chapter

Charles University in Prague
Slovak University of Technology
University of Žilina
Matej Bel University in Banská Bystrica

Czech Technical University in Prague

Technical University of Ostrava
Pavol Jozef Šafárik University in Košice
Masaryk University
University of West Bohemia

CTU Open Contest 2019

Beer Flood System

`flood.c`, `flood.cpp`, `Flood.java`, `flood.py`

Triceratops brewery uses an intricate automated system of pipes to deliver their beer to many local pubs, restaurants and other points of interest. The system is called the Beer flood system. Recently, the brewery is planning its major overhaul.

The system consists of one source station, one collector station, pumping stations and pipes. The source station is located in the brewery. All pumping stations are located in pubs, restaurants, etc. The location of the collector station is somewhat mysterious, however, it is not important for the system function.

The pipes connect some number of stations. The source station is connected to at least one other station and also the collector station is connected to at least one other station. In the extreme case, when the collector station is identical to the source station, there are no pumping stations and no pipes in the system.

The beer flows from the source station through the pipes to other stations where some of it may be (and often is) consumed. Remaining beer which is not consumed in a pumping station automatically continues to flow through the pipes to other pumping station or stations or to the collector station.

The direction of beer flow in the system is always such that the beer which has left a station cannot return to the same station again. A circular flow of beer through any subset of stations never appears in the system. Also, the direction of beer flow between any pair of connected stations is fixed and does not change over time. The size of the beer flow from the source station and through the system is sufficient to supply all pumping stations. There is always some flow of beer through each pipe in the system.

The current Beer flood system was built a long time ago when computer-based optimization was only a dream of the future. Later it became clear that very probably some carefully chosen redundant pipes may be removed from the system and beer flow through the remaining pipes adjusted in such way that the system main features will be preserved. In particular, all pumping stations will still receive a sufficient flow of beer from the source station and all beer unconsumed in pumping stations will still reach the collector. Also, there will be again some flow of beer through each pipe in the system and the direction of the beer flow will not change in any of the remaining pipes. The capacity of all pipes is so big that the remaining pipes need not to be expanded, even though the beer flow through some of them is increased.

Given the topology of the Beer flood system, calculate the maximum number of pipes that can be removed from the system.

Input Specification

The first line contains two integers N, M ($1 \leq N \leq 2000, 0 \leq M \leq 5000$). N is the number of all stations in the Beer flood system, including the source station and the collector station. M is the number of pipes in the system. Stations are labeled by integers $1, 2, \ldots, N$. Next, M

lines follow, each specifies one pipe and the direction of beer flow through it. The line contains two integers X and Y ($1 \leq X, Y \leq N; X \neq Y$), beer flows from station X to station Y. All pipes connect unique pairs of stations, no pipe connects a station to itself. In the input, there is exactly one station that receives no flow, it is the source station. Also, from exactly one station no beer flows to any other station, it is the collector station.

Output Specification

Output a single integer specifying the maximal number of pipes that can be removed from the Beer flood system.

Sample Input 1

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

Output for Sample Input 1

```
2
```

Sample Input 2

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

Output for Sample Input 2

```
0
```