Bridge Collapsing

Objective
Give practice graph searches/distances.
Give practice with and inform students about the concept of the centroid of a tree.

Story
After the completion of the final scroll. You notice some phrases were not translated. You read it out loud to yourself, and the chamber you're in erupts in a cacophony of mechanical noises. The chamber slowly fills with sand and you realize that you might have just cast some sort of self-destruction spell.

You hurry out of the chamber into the open area with the pedestals, many of which have fallen over. The archeologists are running circles on the pedestals across the two-way, narrow pathways trying to reach their ropes. You begrudgingly decide to help them get out. You will tell a group of non panicking archeologists to remove some of the pathways.

Your goal will be to make it so that there are no sequences of distinct pathways that would enable an archeologist walking said sequence to end their movement on the same pedestal where they start from. Additionally, you don't know where the archeologists will start and end at once the redundant pathways are removed, so you want to ensure that the longest distinct pedestal path an archeologist can walk is minimized.

All pedestals will be able to reach each other to begin with. You will also try to avoid indirecty disconnecting any pairs of pedestals. In other words, an archeologist should be able to move from pedestal i to j through a series of moves after removing ALL redundant pathways.

Problem
Given the current state of the pedestals and their pathways, determine the longest distance an archeologist might have to walk under the assumption that the graph will be turned into a graph with no cycles.

Input
Input will begin with a line containing 2 integers, v and e ($1 \le v \le 10{,}000$; $v - 1 \le e \le 10{,}000$), representing the number of pedestals and the number of pathways respectively. The following e lines will each contain 2 integers, the i-th of which are a_i and b_i ($1 \le a_i, b_i \le v$), which represents the start and end pedestal index for the i-th narrow pathway.

The pathways will ensure that all pedestals are connected

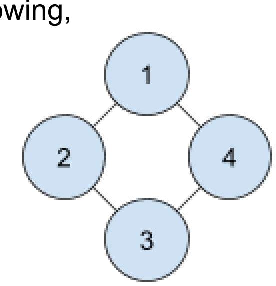
Output
Output should contain 1 line with a single integer representing the longest path required by an archeologist given that no cycles exist within the resulting graph.

Sample Input	Sample Output
4 4 1 2 2 3 3 4 4 1	3
6 7 1 3 3 5 5 1 2 4 4 6 6 2 5 6	3

Explanation

Case 1

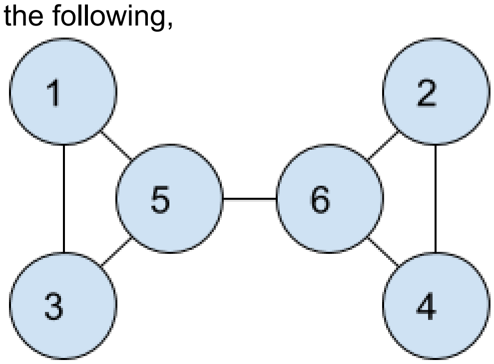
The graph in the first case is the following,



One edge must be removed. Each of them would cause the same graph structure. If we cut the edge from 1 to 4, we have a worst case path of 3 (1,2), (2,3), (3,4).

Case 2

The graph in the first case is the following,



We must cut either (1,3), (1, 5), or (3,5). We must also cut (2,4), (2,6), or (4,6). It is ideal to cut (1,3) and (2,4). The worst case distance is from 1/3 to 2/4. One of those paths would be the 3 edges (1,5), (5,6), and (6,2).

Hints

Graph Type: The graph type is an undirected graph because the pathways are two way. Additionally the graph is connected. We need to remove cycles, which means we need to form a tree.

Tree Diameter: The longest path(s) within a tree is called the diameter. The diameter of the tree can be found by BFS from any node in the tree. The furthest leaf found using the BFS represents one node on some diameter. To find the corresponding leaf node on the diameter, one can BFS from that original furthest leaf. The last node visited in the second BFS will be the other node on the diameter.

Tree Centers: Trees have either 1 or 2 nodes that represent the center of the tree. Every diameter of the tree passes through the center(s) of the tree. Every tree has a center.

Semi-Brute Force: When trying to find the edges to remove, the center of the tree can be bruteforced (try all nodes/connected node pairs).

Diameter Reduction: To reduce the diameter of the tree, a method should choose the paths that make the nodes closest. The node can be made closest by using a BFS. Once the edges are selected. The two BFS approach can be used to find the shortest distance.

Grading Criteria

- Good comments, whitespace, and variable names
  - 15 points
- Brute force the center of the tree
  - 10 points
- Find all the edges using a BFS
  - 10 points
- Find the diameter using 2 BFS's
  - 10 points
- Store the graph in an adjacency list
  - 5 points
- Programs will be tested on <u>10 cases</u>
  - 5 points <u>each</u>

No points will be awarded to programs that do not compile using javac.

Sometime a requested technique will be given, and solutions without the requested technique will have their maximum points total reduced. For this problem use a Breadth First Search (BFS). ***<u>Without this programs will earn at most 50 points!</u>***

Any case that causes a program to return a non-zero return code will be treated as wrong. Additionally, any case that takes longer than the maximum allowed time (the max of {5 times my solutions time, 10 seconds}) will also be treated as wrong.

<u>No partial credit will be awarded for an incorrect case.</u>