**Problem Name:** Minimize malware spread

**Topics:**

**Companies:**

**Level:** Easy

**Language:** C++

**Problem Statement**:

Doge is solving a DSA problem but he is having some difficulty with it. Can you help Doge so that he can complete it and play with his friend Cheem?

Problem is:

**Input Format:**

The first line of input is an integer value n (total no of vertex in the graph).

The Second line of input contains integer value m (total no of edges in the graph)

The next m lines contain two space-separated integers forming a graph.

Last line of input contains the integer value source and destination

**Output Format:**

**Constraints:**

**Examples:**

**Approach one Solution:**

**Explanation:** Union found all nodes.  
Count the union size of each union set.  
Count the malware number of each union set.

Return the biggest union's malware if there is one and only one malware.  
If no such union that has and has only one malware,  
return the malware with minimum index.

**Code:**

**Time Complexity**: O(N^2)

**Space Complexity:**

**Approach second Solution:**

Explanation: Let's imagine that each initial node has a unique malware signature i + 2, where i is the node number, and 2 is just a sentinel value. When we spread the malware (using DFS), we mark all infected nodes in the graph with that signature. We also count infected nodes with that signature and store the number in the sizes array.

Now, when we about to spread the malware from another initial node, it may be already infected. We know the signature, so we can identify the 'zero' node. Since both 'zero' and initial nodes cannot be removed, we set their infected nodes count to 1 in the sizes array.

In the end, our sizes array will contain 1 or more for all initial nodes. We can just return the index of the node with the biggest value.

**Code:**

**Time Complexity**: O(N)

**Space Complexity:** O(N).