**Problem Name:** Number of operation to make a network connected

**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:** Goal: We need to connect all the computers (directly or indirectly). We have to return the minimum number of operations that are required to connect the computers. An operation consists of removing a cable between two directly connected computers and put it between two disconnected computers.

This problem is a simple variation of counting the number of islands.

We know that the minimum number of edges required for a graph with n nodes to remain connected is n - 1. Similarly, if there are k components in a disconnected graph, then we need at least k - 1 edges to connect every component.

With that in our mind, we will start with our **base condition**. If the number of edges in the graph is greater than n - 1 or not. If not, we will return -1.

Next, we will count the number of components (k). As I already mentioned, we will need k - 1 operations to connect the computers (components). And that is our **answer!**

We can also solve this using Union-Find approach and count the number of components.

**Code:**

**Time Complexity**:

**Space Complexity:**

**Approach second Solution:**

Explanation: // Typical union and find problem. We need to find how many 'redundant edge' such that its vertices

    // have the same parent. We also need to find out how many groups of PCs that are not connected.

    // In order to make all PCs connected, we need to make sure:

    // number\_of\_redundant\_edges >= number\_of\_groups - 1.

**Code:**

**Time Complexity**: O(N)

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