Minimum Operations to Make Network Connected

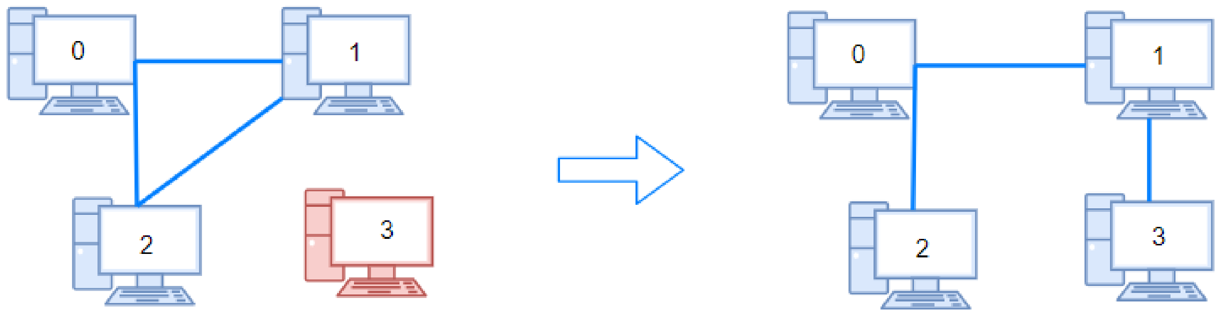
LeetCode

There are n computers numbered from 0 to n - 1 connected by ethernet cables connections forming a network where connections[i] = [a_i, b_i] represents a connection between computers a_i and b_i. Any computer can reach any other computer directly or indirectly through the network.

You are given an initial computer network connections. You can extract certain cables between two directly connected computers, and place them between any pair of disconnected computers to make them directly connected.

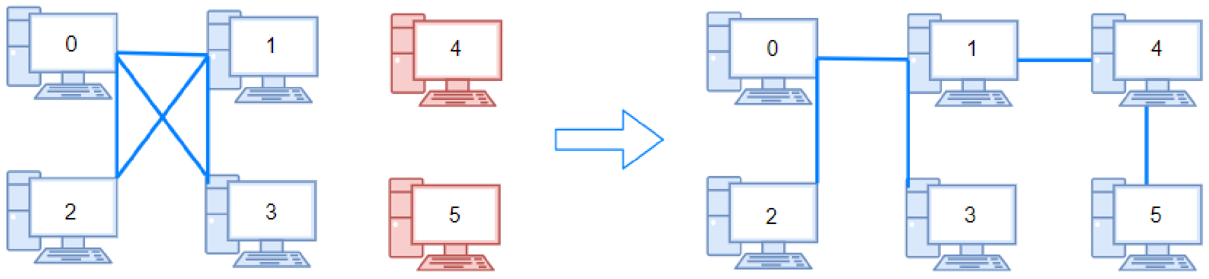
Return *the minimum number of times you need to do this in order to make all the computers connected*. If it is not possible, return -1.

Example:



Output: 1

Example 1:



Output: 2

Approach 1: Function to determine the minimum operations to make the network connected

- **Explanation:**
    - The **makeConnected** function determines the minimum operations needed to make the network connected using the union-find algorithm.
    - It initializes each node as its own parent and keeps track of ranks to optimize union operations.
    - The function iterates over the edges, performing union operations and decrementing the count of connected components.
    - If there are insufficient edges to potentially connect all vertices, the function returns -1.
- **Time Complexity:**
    - **The time complexity is O(E * α(V)), where E is the number of edges, and α(V) is the inverse Ackermann function (effectively constant time for practical purposes).**
        - **The union-find operations dominate the time complexity.**
        - **In practice, α(V) is very small, so we can simplify the time complexity to O(E).**
- **Space Complexity:**
    - **The space complexity is O(V), where V is the number of vertices.**
        - **Storing parent and rank information for each vertex.**