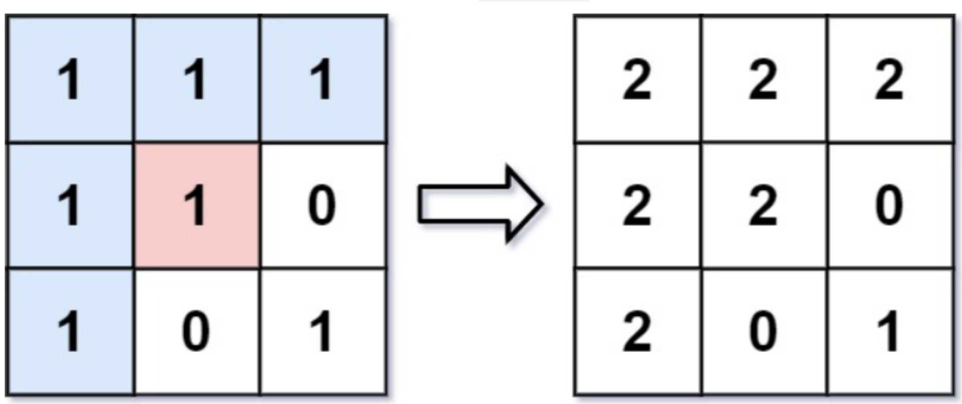
Assignment : Graph -4

Q1 An image is represented by an m x n integer grid image where image[i][j] represents the pixel value of the image.
You are also given three integers sr, sc, and color. You should perform a flood fill on the image starting from the pixel image[sr][sc].

To perform a flood fill, consider the starting pixel, plus any pixels connected 4-directionally to the starting pixel of the same color as the starting pixel, plus any pixels connected 4-directionally to those pixels (also with the same color), and so on. Replace the color of all of the aforementioned pixels with color.

Return the modified image after performing the flood fill.

Example 1:



Input: image = [[1,1,1],[1,1,0],[1,0,1]], sr = 1, sc = 1, color = 2
Output: [[2,2,2],[2,2,0],[2,0,1]]

Explanation: From the center of the image with position (sr, sc) = (1, 1) (i.e., the red pixel), all pixels connected by a path of the same color as the starting pixel (i.e., the blue pixels) are colored with the new color.
Note the bottom corner is not colored 2, because it is not 4-directionally connected to the starting pixel.

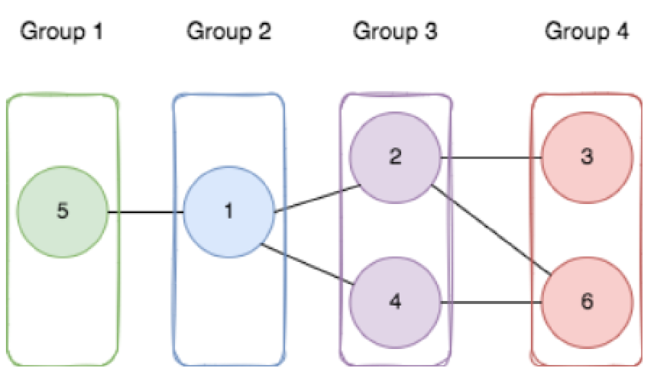
Q2 You are given a positive integer n representing the number of nodes in an undirected graph. The nodes are labeled from 1 to n.
You are also given a 2D integer array edges, where edges[i] = [ai, bi] indicates that there is a bidirectional edge between nodes ai and bi. Notice that the given graph may be disconnected.

Divide the nodes of the graph into m groups (1-indexed) such that:

- Each node in the graph belongs to exactly one group.
- For every pair of nodes in the graph that are connected by an edge [ai, bi], if ai belongs to the group with index x, and bi belongs to the group with index y, then |y - x| = 1.

Return the maximum number of groups (i.e., maximum m) into which you can divide the nodes.
Return -1 if it is impossible to group the nodes with the given conditions

Example 1:



Input: n = 6, edges = [[1,2],[1,4],[1,5],[2,6],[2,3],[4,6]]
Output: 4

Explanation: As shown in the image we:
- Add node 5 to the first group.
- Add node 1 to the second group.
- Add nodes 2 and 4 to the third group.
- Add nodes 3 and 6 to the fourth group.
We can see that every edge is satisfied.
It can be shown that that if we create a fifth group and move any node from the third or fourth group to it, at least on of the edges will not be satisfied.

Q3 There is a bi-directional graph with n vertices, where each vertex is labeled from 0 to n - 1. The edges in the graph are represented by a given 2D integer array edges, where edges[i] = [ui, vi] denotes an edge between vertex ui and vertex vi. Every vertex pair is connected by at most one edge, and no vertex has an edge to itself.

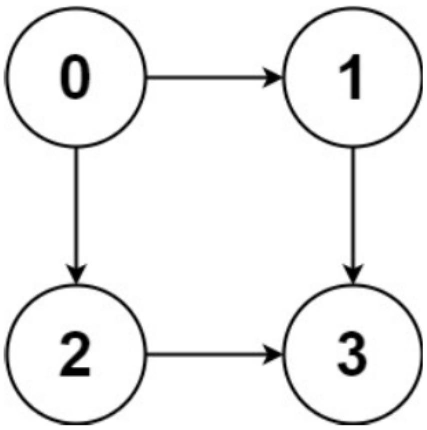
Return the length of the shortest cycle in the graph. If no cycle exists, return -1.

A cycle is a path that starts and ends at the same node, and each edge in the path is used only once.

Q5 Given a directed acyclic graph (DAG) of n nodes labeled from 0 to n - 1, find all possible paths from node 0 to node n - 1 and return them in any order.

The graph is given as follows: graph[i] is a list of all nodes you can visit from node i (i.e., there is a directed edge from node i to node graph[i][j]).

Example 1:



Input: graph = [[1,2],[3],[3],[]]
Output: [[0,1,3],[0,2,3]]

Explanation: There are two paths: 0 -> 1 -> 3 and 0 -> 2 -> 3.