**Data Structures and Algorithms Lab**

**Lab Test - V2 (CO4)**

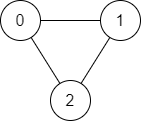
**Maximum Marks: 20 Max Time: 40 minutes**

Set – A

There is a graph with n vertices, where each vertex is labeled from 0 to n - 1 (inclusive). The edges in the graph are represented as a 2D integer array edges, where each 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.

You want to determine if there is a valid path that exists from vertex source to vertex destination.

Given edges and the integers n, source, and destination, return true *if there is a valid path from* source *to* destination*, or* false *otherwise.*

**Example 1:**

Input: n = 3, edges = [[0,1],[1,2],[2,0]], source = 0, destination = 2

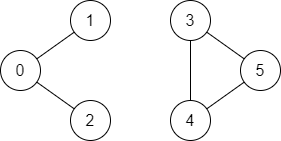
Output: true

Explanation: There are two paths from vertex 0 to vertex 2:

- 0 → 1 → 2

- 0 → 2

**Example 2:**



Input: n = 6, edges = [[0,1],[0,2],[3,5],[5,4],[4,3]], source = 0, destination = 5

Output: false

Explanation: There is no path from vertex 0 to vertex 5.

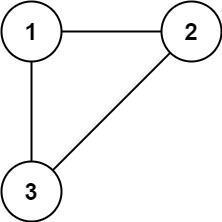
Set – B

A tree can be considered as an **undirected graph** that is connected and has no cycles.

You are given a graph that started as a tree with n nodes labeled from 1 to n, with one additional edge added. The added edge has two **different** vertices chosen from 1 to n, and was not an edge that already existed. The graph is represented as an array edges of length n where edges[i] = [ai, bi] indicates that there is an edge between nodes ai and bi in the graph.

Return *an edge that can be removed so that the resulting graph is a tree of*n*nodes*. If there are multiple answers, return the answer that occurs last in the input.

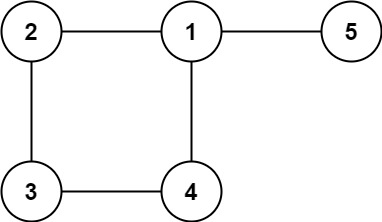
**Example 1:**



**Input:** edges = [[1,2],[1,3],[2,3]]

**Output:** [2,3]

**Example 2:**



**Input:** edges = [[1,2],[2,3],[3,4],[1,4],[1,5]]

**Output:** [1,4]