Final Assignment (30 marks)

Deadline: On or before 2nd July

Task Overview

Implement the following FIVE functions and test their correctness in the `main` function. Assume the input graph G is simple, undirected, unweighted and represented using an **adjacency matrix**. Also, assume vertices are labeled as integers starting from 0.

Reuse one function as a subroutine within another when appropriate.

1. **Test whether a graph is a tree**

   ```
   boolean isTree(Graph G)
   ```

   Returns `true` if the graph G is a tree (i.e., connected and acyclic), otherwise returns `false`.

2. **Test whether a graph is 2-colorable**

   ```
   boolean isTwoColorable(Graph G)
   ```

   Returns `true` if the graph G can be colored using two colors such that no two adjacent vertices share the same color, otherwise returns `false`.

3. **Test whether a graph is bipartite**

   ```
   boolean isBipartite(Graph G)
   ```

   Returns `true` if the graph G is *bipartite*.
   Note: A graph is *bipartite* if its vertex set can be partitioned into two disjoint subsets such that no edge connects vertices within the same subset.

4. **Find the diameter of a graph**

   ```
   int diameter(Graph G)
   ```

   Returns the *diameter* of the graph G, which is the length of the longest shortest path between all pair of vertices. (Among the shortest paths between every pair of vertices in the Graph, find the length of the longest path.)

5. **Identify the Most Critical vertex for Cascading Failure**

In complex systems such as electric grids, a *cascading failure* occurs when the failure of one or a few components triggers a chain reaction, progressively leading to the failure of other components. This often happens due to positive feedback and interdependence between parts of the system.

In this task, an electric grid is modeled as a graph where:

- Each vertex represents a power infrastructure component (e.g., power station, substation).
- Each edge represents a potential path through which failure can propagate from one vertex to another.

Write a function `int mostCriticalVertex(Graph G)` to determine the most critical vertex in the network — that is, the vertex whose initial failure leads to the complete failure of the entire network in the shortest number of propagation steps.

Instruction:

Complete all the required functions as described in this assignment. Your evaluation will be based on a viva conducted on your submitted code. You are required to attend the viva in my office during my *counseling hours*. Please notify me in advance before your visit.