Loyola University Chicago
Department of Computer Science
COMP 272: Data Structures II (Spring 2024)

Assignment # 8

This is an individual assignment.
Deadline: Tuesday**, Monday 29th, 2024, 11:55PM.**
You need to submit your solutions in Repl.it.
Before submission, make sure your code passes all the JUnit tests.

Graph Traversal (33 points)

You are to implement the findRoot method in the Graph class. This method should determine the "root" of the graph, defined as the vertex with no incoming edges. The method should return the ‘value’ of the root vertex. If there is no such unique vertex, or more than one root vertex is found, then return -1.

Master Programmer (33 points)

To reach your goal of 'master programmer', you need to complete *'n'* certification exams, each being specific to a topic. Some exams have prerequisites of needing to take and pass earlier certificate exams.

You can represent these *'n'* exams as nodes in a graph, numbered from 0 to *n*-1. And represent the prerequisite between taking exams as directed edges between two nodes which represent those two exams.

You are given a 2-dimensional array of exam prerequisites where prerequisites[*i*] = [*ai*, *bi*] indicating that you must take exam *bi* first if you want to take exam *ai*. For example, the pair [0, 1], indicates that to take exam certification 0, you have to first have the certification for exam 1.

Return true if you can finish all certification exams. Otherwise, return false (e.g., meaning it is a cyclic or cycle graph).

Example 1:

Input: numExams = 2, prerequisites = [[1,0]]

Output: true

Explanation: There are a total of 2 exams to take. To take exam 1 you should have completed the certification of exam 0. So, it is possible (no cyclic or cycle graph of prereqs).

Example 2:

Input: numExams = 2, prerequisites = [[1,0],[0,1]]

Output: false

Explanation: There are a total of 2 exams to take. To take exam 1 you should have completed the certification of exam 0, and to take exams 0 you should also have completed the certification of exam 1. So, it is impossible (it is a cycle graph).

Number of Groups (33 points)

There are n people. Some of them are connected as friends forming a group. If person 'a' is connected to person 'b', and person 'b' is connected to person 'c', they form a connected group.

Not all groups are interconnected, meaning there can be 1 or more groups depending on how people are connected.

This example can be viewed as a graph problem, where people are represented as nodes, and edges between them represent people being connected. In this problem, we are representing this graph externally as an non-directed Adjacency Matrix. And the graph itself may not be fully connected, it can have 1 or more non-connected compoents (subgraphs).

Example 1:

Input : Adjacency Matrix = [[0,1,0], [1,0,0], [0,0,0]]

Output: 2

Explanation: The Adjacency Matrix defines an undirected graph of 3 nodes (indexed 0 to 2). Where nodes 0 and 1 aee connected, and node 2 is NOT connected. This forms two groups of nodes.

Example 2:

Input : Adjacency Matrix = [[0,0,0], [0,0,0], [0,0,0]]

Output: 3

Explanation: The Adjacency Matrix defines an undirected graph of 3 nodes (indexed 0 to 2). There are no connected nodes, hence forming three groups.

Example 3:

Input : Adjacency Matrix = [[0,1,0], [1,0,0], [0,1,0]]

Output: 1

Explanation, The adjacency Matrix defined an undirected graph of 3 nodes (index 0 to 2). All three nodes are connected by at least one edge. So they form on large group.

Submission:

1) Assignment is worth 100 points, since the 3 sections add up to 99, you get 1 point for free ☺
2) **Before submission, make sure your code passes all the JUnit tests.** Keep in mind, however, that passing the test cases does not guarantee that your code is correct or efficient. Your assignment will be graded considering test results, correctness, and efficiency.
3) The submission should be completed in Replit.
4) Include your name and class number as comments at the top of each submitted Java file.