Assignment 03: Graph Traversals

In this assignment, you will implement a graph using the adjacency matrix (as shown in class), and traverse through the graph to determine if a path exists from one vertex to another.

## The input files

You will be given two types of input files.

### The Graph Input File

The first input file will contain an adjacency matrix representing a directed graph in the following format:

0 0 0 1 0

0 0 1 0 1

1 0 0 0 0

0 1 0 0 1

0 0 0 0 0

The first line of the input file will be the number of vertices in the graph, immediately followed by the adjacency matrix for the graph in the format shown above.

### The Vertices Input File

The second type of input file will contain records consisting of a pair of vertices.

The format of the second file is start\_node end\_node. For example:

0 1

4 2

## The Program

### The Driver class:

The driver class must read from the graph input file and create a Graph object (implemented using the adjacency matrix).

Then, it should read from the vertices input file and determine if there is a path between these two vertices in the graph and to print the results to the screen (Sample output is shown below. Your output should match that exactly.)

### The Graph class:

You must program the graph data structure from scratch (as shown in class, you may use all code given in class); do not use any classes from the Java libraries to implement the graph. Your graph class must contain a minimum (additional methods are allowed) of two methods:

* addEdge – add the edge to the adjacency matrix
* checkEdge – take in two vertices (see sample output below) and print the output for whether a path exists between the vertices

## Sample Output

### Using graph1.txt and check1.txt

The path from 0 to 1 is: 0 - 3 - 1

No path from 4 to 2 exists.

The path from 0 to 4 is: 0 - 3 - 1 - 4

An direct edge between 1 and 4 exists.

The path from 2 to 4 is: 2 - 0 - 3 - 1 - 4

The path from 3 to 0 is: 3 - 1 - 2 – 0

### Using graph2.txt and check2.txt

No path from 0 to 1 exists.

The path from 4 to 2 is: 4 - 6 - 3 - 2

No path from 7 to 4 exists.

The path from 1 to 4 is: 1 - 2 - 6 - 3 - 0 - 4

The path from 6 to 4 is: 6 - 3 - 0 - 4

An direct edge between 3 and 0 exists.

No path from 7 to 1 exists.

The path from 5 to 7 is: 5 - 1 - 2 - 6 - 7

No path from 6 to 1 exists.

The path from 5 to 2 is: 5 - 1 - 2

No path from 2 to 5 exists.

## Submission and Collaboration

This program must be **completed on your own**. **You should name your program YourLastNameAssign03, zip the entire project folder, and turn in the zipped folder to Blackboard**. **Programs that do not give the expected sample output for both files provided on Blackboard will automatically receive a 50% deduction in grade**. Further deduction may occur depending on how close it was to the expected output. **Programs that are not turned in correctly or do not compile will receive a zero.**