COMP 482 Project 2

Background: Graphs and graph algorithms are extremely important in our everyday lives. From your GPS giving you the best route possible to keeping track of players in a video game, graphs have tons of applications.

Objective: This time we are going to have 2 objectives for the program you are going to create utilizing an adjacency list to represent a graph:

- Objective #1 – report if a cycle exists in the graph (there will either be 1 cycle, or no cycles).
- Objective #2 – **Starting from Node 1**, Use BFS to tell me how many layers exist in the given graph. The root will be layer 0.

Input Format: You must create a file called “input2.txt”. The input file is going to contain an adjacency matrix only. (see the examples below for what it will look like).

EXAMPLE #1: Given an input2.txt file of:

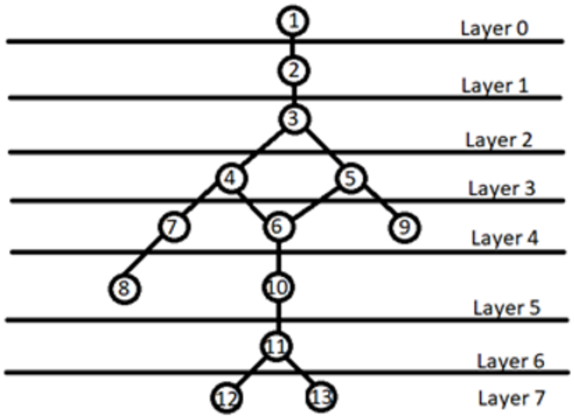
```
0 1 1
1 0 1
1 1 0
```

Your output should be:

```
Does cycle exist?: Yes
Number of layers in the graph: 1
```

Example #2: Given an input2.txt file of:

```
0 1 0 0 0 0 0 0 0 0 0 0 0
1 0 1 0 0 0 0 0 0 0 0 0 0
0 1 0 1 1 0 0 0 0 0 0 0 0
0 0 1 0 0 1 1 0 0 0 0 0 0
0 0 1 0 0 1 0 0 1 0 0 0 0
0 0 0 1 1 0 0 0 0 1 0 0 0
0 0 0 1 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 1 0 1 1
0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0
```



Your output should be:

```
Does a cycle exist?: Yes
Number of layers in the graph: 7
```

Important Note: This project will be hard to visualize just using the adjacency matrix. It will ABSOLUTELY help you to draw these graphs from the given adjacency list. I included a sample drawing of example 2 above! Remember: A “0” means there is no connection to a node and “1” means there is a connection to a node!

Project information and Project Submissions:

- Projects will be done **ONLY** in **Java** (No other languages will be accepted)
- Students should begin to work on projects when the project specifications are released.
- Projects will be released as early as possible to students, and you are encouraged to complete the projects as early as possible. Even if a topic has yet to be covered, if students have taken the time to learn the material beforehand, feel free to attempt projects early and submit them early.
- You will be able to submit your project as many times as you wish until the deadline given. (**For a regrade you must submit the project at least 7 days prior to the deadline otherwise there will be no regrade**)
- **Late projects will not be accepted.**

Projects must be submitted as follows:

- You must submit your project to Canvas **ONLY.** Email submissions will not be accepted.
- The file must be submitted in a ".zip" format
- The ".zip" file should be named with your First and Last name with the project number at the end (Example: DinoBiel1.zip)
- If working in visual studio please do not zip the entire project. Only zip the ".java" file (doing this will cost you points!)
- you must use the DEFAULT package, if not sure how to do this, ask the professor.
- Failure to comply with the rules above will result in a major loss of points on the project!

Grading Rubric: When grading your projects I will assign grades based on the following criteria:

- Does the project work according to the specification including reasonable time complexity? – **50%**
- Does the project utilize the concepts requested in the specification? – **30%**
- Is the code provided in the project well formatted? – **10%**
- Does the code contain sufficient and useful comments to explain sections of the code? – **10%**