Lab 9: Cycle detection

1. Introduction

In the lecture today we talked about two graph traversals, BFS and DFS. In this lab you will first implement Graph and Vertex class to represent directed graphs in your program, and then you will implement a method in Graph class that determines if there exists a cycle in this directed graph.

The code skeleton is under: **/comp/15/files/l9**

1. Log in to the CS homework server.
2. Move to your **comp15** directory that you created in Lab 1.
3. Create a directory named **lab9** under the comp15 directory.
4. Move to the lab directory.
5. Copy the code skeleton to the current working directory.
6. Leverage the features of Git to manage your progress toward writing the program.

The code skeleton provides you with five files: **test.cpp**, **Graph.hpp**, **Graph.cpp**, **Vertex.hpp**, **Vertex.cpp**.

- Note that the copy constructor and assignment operator of the Graph class and of the Vertex class are commented out, so that this lab can be completed within our lab time. However, you are encouraged to work on them later on.
- **test.cpp** provides you with two tests with assert() that your implementation is supposed to pass.

2. Requirements

1. Implement the Vertex class.
    a. You can use the vector, which is one of C++ STL, to hold its adjacent vertices.
    b. The default constructor: You will define what should happen when the default constructor is used.
    c. The addEdge() and getAdjacentVertexAt(): You will define what should happen when the given information is not appropriate.
2. Implement the Graph class.
    a. You can use the vector, which is one of C++ STL, to hold all vertices.
    b. Note: isCyclic() returns true if there is a cycle in the graph, and false otherwise.
3. Write at least five additional test cases in **test.cpp**.
4. Pass the tests, including yours, with assert() written in test.cpp with no memory leaks and no memory errors.

3. README

Create the README file that includes the following categories with appropriate section headers.

1. **Name**: Your name.
2. **Date**: The last updated date.
3. **Summary**: A brief summary of the lab.
4. **Files**: A list of files that are necessary to build and test the program.
5. **Instructions**: A sequence of commands to compile and test the program. Note that you are expected to report procedures without using the make command.
6. **References**: A list of citations to information used to complete the lab.

4. Submission

Submit your files listed below using Gradescope.
Files: **test.cpp Graph.hpp Graph.cpp Vertex.hpp Vertex.cpp README**