CSCI 230 -- PA 9

Graph Representations & Traversals

Feel free to discuss and help each other out but does not imply that you can give away your code or your answers! Make sure to read all instructions before attempting this lab.

You can work with a lab partner and each one must submit the same PDF file (include both names in the submission file). Each person must include a brief statement about your contribution to this assignment.

You must use an appropriate provided template from Canvas and output "Author: Your Name(s)" for all your programs. If you are modifying an existing program, use "Modified by: Your Name(s)".

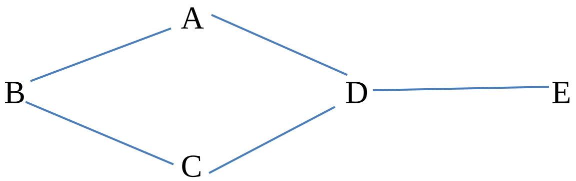
Exercise 1: Set up your own code to represent an undirected graph using Adjacency List (use string for vertex and int for edge. Try the test case below first and then create a simple graph with 4 vertices and 6 edges and print it. Output your graph in this format:

```
Vertex A
 2 adjacencies:(C, 200)  (B, 100)
Vertex B
 1 adjacencies:(A, 100)
Vertex C
 1 adjacencies:(A, 200)
```

For C++, you can start with my AdjacencyMapGraph class to use a list instead of map to create AdjacencyListGraph class (Adjacency List option). Files available: Graph.h, AdjacencyListGraph.h, and testGraph.cpp.

For Java, you can modify AdjacencyMapGraph to use a list instead of map to create AdjacencyListGraph class (Adjacency List option). Files to use: AdjacencyListGraph.java and GraphExample.java.

Exercise 2: Implement either **DFS** or **BFS** using your graph class from exercise 1. You can set up DFS(G, v) like the book (perform DFS on a graph) or G.DFS(v) where DFS() is a member of Graph class (use similar set up for BFS). For C++, you can use Decorator.h and DFS.h and use GaphAlgorithms.java for Java. Print out the vertices and discovery/forward edges in the order that they were visited (should be vertex, discovery edge, vertex, etc.). Try the following graph and start out with vertex A:



Question 1: Discuss advantages of Adjacency List over Edge List Structure for an undirected graph.

Question 2: Does DFS or BFS guarantee to visit the vertices in a certain order? Why or why not?

Extra Credit: Set up code to represent an undirected graph using either a simple matrix or adjacency matrix. There is no code for it, but it should be straight forward with a simple matrix that stores only 0 or 1. Try it out with the same two test cases in exercise 1.

Fill out and turn in the PA submission file for this assignment (save as PDF format).