Homework: **Graph Theory and Graph ADTs**

1. Check the Assignment Schedule for the DUE date.
2. Submit via Moodle.

Problem 1:

A customer asks you to provide the most efficient implementation of the following operations over a large graph (about 10,000 vertices): add/delete edges/vertices.

1. What questions would you ask the customer that will help you to make the right decision about the customer's request? [Hints: Check the types of graphs]
2. Provide three scenarios depending on the answers to those questions.
3. For each scenario, propose the most efficient data structure and supply the time complexity of the requested operations in terms of big-O. What is the space requirement for the selected data structure?

Problem 2:

Let G be a simple directed graph with N nodes. The path matrix of G is the n-square matrix P defined as follows: Each (i,j) element of the matrix is 1 if there is a path from node i to node j in G.

1. Using only an adjacency matrix A of graph G defined as a boolean (0,1) matrix, design an algorithm that constructs the matrix P in the most efficient manner. Write any mathematical relationship for P you will use for your algorithm and justify this relationship (formal or informal proof is up to you). [Hint: What is A times A? What is A times A times A?]
2. Provide a pseudo-code of the algorithm.
3. Analyse the time complexity of your algorithm in terms of big-O.
4. Define the values of N, for which this algorithm becomes impractical (e.g., will take more than a day to get an answer) on my laptop running at 2.5 GHz (i.e., 2.4 * 10^9 floating point operations per second).
5. If my laptop has 512 MB, 1GB, or 16 GB of RAM, will your answer to (4) change given my memory constraints (i.e., in terms of big-O for space requirements) and how many bytes do you require per your selected data structure; crude estimation with the justification is enough?