CSC 225 (Fall 2016): Algorithms and Data Structures I
Assignment 4—theoretical part

Note: your solution must be submitted as a **single pdf file**

Question 1. [Graphs]

Let G be the undirected graph with vertices $V = \{0,1,2,3,4,5,6,7,8\}$ and edges $E = \{\{0,1\}, \{0,4\}, \{0,5\}, \{1,2\}, \{1,5\}, \{1,6\}, \{2,3\}, \{2,6\}, \{3,6\}, \{3,7\}, \{4,8\}, \{5,6\}, \{5,8\}, \{6,8\}, \{7,8\}\}$

(a) Draw G in such a way that no two edges cross (A graph that can be depicted without edge crossing is called a *planar graph*.)

(b) Show an adjacency list representation of G.

(c) What is the adjacency matrix representation of G? Show the adjacency matrix where a 1 represents the existence of an edge and 0 denotes non-adjacent vertex pairs.

(d) For the graph G in Question 1, assume that in a traversal of G, the adjacent vertices of a given vertex are returned in their numeric order.

(i) Order the vertices as they are visited in a DFS traversal starting at vertex 0.

(ii) Order the vertices as they are visited in a BFS traversal starting at vertex 0.

Question 2. [Graph Properties]

Let undirected simple graph $F = (V, E)$ be a forest with n vertices, m edges and k connected components. *Prove* that the number of edges in F is $m = n - k$.

Question 3. [Topological Sort]

(a) In pseudocode, design an algorithm that determines whether a digraph has a *unique* topological ordering. Your algorithm should return the ordering if a unique one exists, and indicate that no unique topological order, or no topological order exists, otherwise.

(b) Are the following statements true or false? Argue convincingly.

(i) A postorder traversal on a tree always produces a topological ordering. For this, assume that consider the tree as directed graph: there is a directed edge (child, parent) between each child and parent [that is, the child is the source, and the parent the destination of the directed edge].

(ii) If a graph has a topological ordering, then a depth-first traversal of the same directed graph will not see any back edges.

Question 4. [Describing problems as computational problems**]**

Consider a social network. The goal is to find in the in the network a largest group of people who are all friends with each other. Describe the problem as a graph problem. Clearly state input and output. Explain why solving your graph problem will solve the original problem.