1. Consider the graph in Figure 1.

   (a) Show the adjacency matrix representation of the graph.
   (b) Show the adjacency list representation of the graph.
   (c) Traverse the graph with DFS and show both the DFS tree and the order the nodes are visited. Assume the edges of the adjacency lists are in increasing order.
   (d) Traverse the graph with BFS and show both the BFS tree and the order the nodes are visited. Assume the edges of the adjacency lists are in increasing order.

![Figure 1: Graph for Question 1.](image)

2. Consider the undirected, unweighted, connected graph with 7 nodes (A, B, C, D, E, F, G) and the following 11 edges:

   - A-B
   - A-C
   - A-F
   - B-C
   - B-D
   - B-G
   - C-E
   - C-G
   - D-E
   - D-F
   - E-F
3. Design an $O(n)$ algorithm to find the length of the longest path in a binary tree (that is, root to leaf). Assume the length of a path is the number of edges.

CSCI 433 students must answer ONE of the following two problems. CSCI 502 students must answer both.

4. Write an algorithm to reverse a linked list. You may modify only the next fields of the nodes. What is the worst-case running time of your algorithm? Be sure to argue that it is correct.

5. Write a $\Theta(n + m)$ algorithm that prints the indegree and outdegree of every vertex in a $m$-edge, $n$-vertex directed graph. Assume the graph is represented using adjacency lists. Argue the algorithm is correct and analyze its complexity.