Instructions: Turn in your answers to the questions below in class on or before the due date. You must include at least a brief summary of each question below as part of your answer (including the question number). Be sure your answers are legible. Note that each question is worth 1 point (for a total of 10 points).

1. Using the Breadth-First Search algorithm from class, give a trace of the algorithm for the following directed graph starting at vertex 0. Assume out-nodes are returned in ascending order. Your trace should include the values of the visited list, the queue as vertices are found, and the search tree as parent-child relationships are established.

2. Give an appropriate (outer) loop invariant to show that the Breadth-First Search algorithm from class finds all vertices that are reachable from a start vertex $s$. Note you can make assumptions about the correctness of the inner loop. Provide a justification for your loop invariant in terms of initialization, maintenance, and termination.

3. Using the connected components algorithm outlined in class, similar to Question 1, give a trace of the algorithm for the graph below.

4. Define a stack-based iterative version of the Depth-First Search algorithm presented in class.

5. Using your algorithm in Question 4, give a trace of the algorithm for the graph in Question 1 starting at vertex 0. Assume out-nodes are returned in ascending order. Your trace should include the values of the visited list, the stack as vertices are found, and the search tree as parent-child relationships are established.
6. Give an appropriate (outer) loop invariant to show that your iterative Depth-First Search algorithm from class finds all vertices that are reachable from a start vertex \( s \). Note you can make assumptions about the correctness of the inner loop. Provide a justification for your loop invariant in terms of initialization, maintenance, and termination.

7. Give a 2-coloring of the following graph.

8. Similar to a 2-coloring, a 3-coloring of a graph has each node assigned one of three colors such that no adjacent graphs have the same color. Give an example of a graph that doesn’t have a 3-coloring.

9. Define an algorithm based on the approach described in class that uses (iterative) Depth-First Search to find a topological sort of a directed graph.

10. Trace your algorithm in Question 9 for the graph in Question 1. Your trace should include the visited list, the stack, and the list (for returning the topological sort) as each node is discovered in the graph.