CMPS 101
Midterm 2 Review Problems

Figure 1:
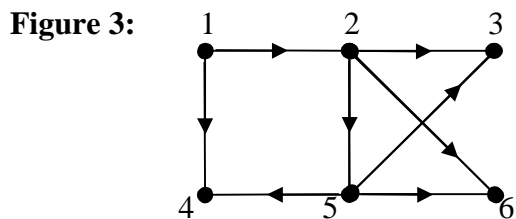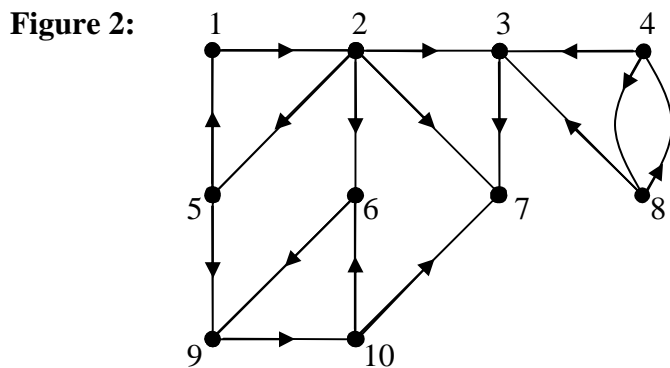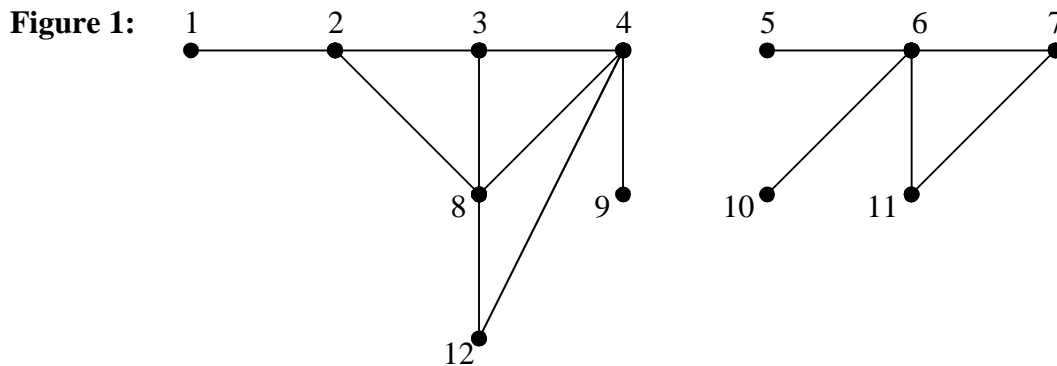
Figure 2:

Figure 3:

Problems

1. Trace BFS on the following graphs. For each vertex, record its color, parent, and distance fields, and draw the resulting BFS tree. Process adjacency lists in (ascending) numerical order.
   a. The graph in figure 1, with 1 as the source.
   b. The directed graph in figure 2 with 1 as source.

2. Trace DFS on the following graphs. For each vertex, record its color, parent, discover, and finish fields, and draw the resulting DFS forest. Classify each edge as tree, back forward, or cross. Process adjacency lists in (ascending) numerical order.
   a. The graph in figure 1. Process vertices in the main loop of DFS in (ascending) numerical order.
   b. The graph in figure 2. Process vertices in the main loop of DFS in (ascending) numerical order.
   c. The transpose of the graph in figure 2. Process vertices in the main loop of DFS in order of descending finish times from part b. Determine the strongly connected components of the graph in figure 2, and draw its component graph. Also determine a topological sort of the strong components.
   d. The graph in figure 3. Process vertices in the main loop of DFS in (ascending) numerical order. Show that this graph is acyclic and determine a topological sort of the vertices.
   e. The graph in figure 3. Process vertices in the main loop of DFS in descending order. Determine a topological sort of the vertices which is different from that in part d.

3. Write an algorithm called isBipartite(*G*) which takes as input a connected (undirected) graph *G* and returns *true* or *false* according to whether or not *G* is bipartite. Hint: see the solutions to hw assignment #5 problem 2 (the wrestler problem).

4. Let $G = (V, E)$ be a connected (undirected) graph. Prove $|E| \geq |V| - 1$. Hint: use induction on $|E|$. Observe that this is Lemma 2 from page 3 of the Graph Theory handout.

5. Let *G* be a directed graph. Determine whether, at any point during a Depth First Search of *G*, there can exist an edge of the following kind.
   a. A tree edge which joins a white vertex to a gray vertex.
   b. A back edge which joins a black vertex to a white vertex.
   c. A forward edge which joins a gray vertex to a black vertex.
   d. A cross edge which joins a black vertex to a gray vertex.

6. a. State the parenthesis theorem.
   b. State the white path theorem.

7. Let *G* be a directed graph. Prove that if *G* contains a directed cycle, then *G* contains a back edge. (Hint: use the white path theorem.)

8. Let *T* be a binary tree, and let $n(T)$ and $h(T)$ denote its number of nodes and height, respectively. Show that $h(T) \geq \lfloor \lg(n(T)) \rfloor$. (Hint: this was proved in the solutions to hw7.)

9. (p.132: 6.2-5)
The code for Max-Heapify is quite efficient in terms of constant factors, except possibly for the recursive call in line 10, which might cause some compilers to produce inefficient code. Write an efficient Max-Heapify that uses an iterative control construct (a loop) instead of recursion.