CIS263 Graph Assignment One

Dr. Denton Bobeldyk

1. Hand draw or digitally render a directed acyclic graph with at least 12 vertices and 12+ edges. Submit an image of this graph.
2. Create a graph data structure using one of the approved programming languages. Store the graph that you created in step one into your data structure.
3. Write a method that performs depth first search and breadth first search. Demonstrate the correctness of each method by searching for at least 2 nodes that are in the graph and 1 that isn't (6 total queries). Submit a screenshot of your program execution.
4. Write a program to perform a topological sort on the graph provided in step one.
5. Hand draw or digitally render the output of your topological sort (draw the graph in a single line going from left to right) and add in each of the directed edges to illustrate the correctness of your top sort algorithm. Submit this image.
6. Hand draw or digitally render a directed graph with at least 12 vertices and 3 strongly connected components. Submit an image of this graph.
7. Programmatically implement an algorithm that finds the strongly connected components in the graph created in step 6 and output each of them (you can just list the vertices for each strongly connected component).

Hand-in:

1. The code used to complete the above tasks (no zip files).
2. A screenshot/image capture for steps 1, 3, 5, 6
3. Demonstration of the functionality of your topological sort method/program
4. Demonstration of the functionality of your strongly connected method/program

Approved programming languages: C, C++, C#, Python, Java.

See blackboard for point breakdown.