Graphs and Breadth-first Search

Homework #10

By

John Kiyak

CS303

November 6 2018

**Problem Specification**

The goal of this assignment was to write a program that would implement the Breadth First Search algorithm and also make an undirected graphs based on the input file that was given to us.

**Program Design**

For my program design, the main function reads and stores data from the ‘mediumG.txt’ or the ‘largeG.txt” file. It reads the text file and stores the data into an array, leaving off the first two data, which were used to find the max value and the number of vertices in the table. Then, I would have a function that would create a graph using the max value and the number of vertices. After creating the graph I would use the BFS algorithm to traverse through the graph that I made.

**Testing Plan**

First, I would maek sure that the input files were being read in, and correctly being output into the arrays. Then, I would make sure that the graph was being created with the correct number of vertices and values from the max value from the input file. Then, I would check that the BFS algorithm was displaying correctly, and had the accurate values for a sample size that I deemed fit.

**Analysis and Conclusions**

For the time complexity of BFS, the normal time would be O(V+E), where V is the number of vertices in the graph, and E is the number of edges in the graph. In our smaller sample size, the mediumG.txt file, there were 250 vertices, with 1273 edges, so the time complexity of the smaller input file would be O(1523). On the other hand, the larger text file, largeG.txt, has 1,000,000 vertices, and 7586063 edges, so it would require a runtime of O(1,000,000 + 7,586,063), which would equal O(8586063). However, in using this, both of these would have a constant time, so no matter how many you get, it would run in constant time. However, the E may vary between constant time and V^2, depending on how big or small the input graph is, and how many connections each vertex takes. So, depending on how big the input graph is, you could have anywhere between constant time and O(n^2) time.