**CS303 Lab 12 – Depth First Search**

**Problem Specification:**

For this lab we have to use LinkedLists to create a Depth First Search program. The program takes in two input files, tinyDG.txt and mediumG.txt and runs depth first search on it. We are also given the task of printing the path from the source to any given vertex and the topological sort.

**Program Design:**

For this program, we had two functions. The first was the DFSearch, which is what I named the function, the idea is that it is the Depth First Search function. This function takes in a graph which I translated to an Array of LinkedLists of vertices. The goal of the function is to essentially go through each vertex and its neighbors set its values, such as the parent, distance, and color. The second function is to print out the vertices and all of its connected neighbors/vertices. When writing my code, I eliminated the color and distance and replaced it with a Boolean value and a integer variable called accessed. Personally, I chose to use a Boolean value because it eliminated all of the need to check for color and it made it simpler for me to understand. I created the accessed variable because I needed to check how many times each vertex had been accessed to make sure that after it had been accessed once, it wouldn’t get changed ever again.

**Testing Plan:**

To test this, I read in both files, mediumG.txt and largeG.txt and create a vertex out of every number in the file and add it to a LinkedList array. This array is an array of LinkedLists and each index represents one of the vertices that is found in the text file. After adding all connected edges to a vertex to its corresponding array index, I run Breadth First Search with that array and a user input integer which is considered the source.

**Test Cases:**

For test cases, I have put in a screen shot of a result of how long it took for BFSearch to go through the LinkedList.

**Text

Description automatically generated**

**Text

Description automatically generatedText

Description automatically generated**

🡨This is a sample path screen shot, it shows the path from the source which is the number on the left of the colon and the edges which are the numbers to the right of the colon.

**Analysis/Conclusion:**

I am confused with my results because somehow, the time that it took for the largeG text file was greater, but not by so much that it would make sense. The mediumG file took approx. 0.0008 seconds while the largeG file took approx. 0.001 seconds. Technically this is a big difference, but I expected the difference to be much greater when looking at the size difference between each file.

**References**:

Text

Description automatically generatedText

Description automatically generatedThe only references I must make are to the pseudocode and my previous lab in which I used a similar format for writing this report. Below I have added images of all my code for this lab.

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated