**.IT 114**

**Programming Assignment 4**

**Visualization of depth first traversal**

**1. Formulating the Problem**

**1.1 Problem Description**

This assignment asks us to create a program that visualizes the solving of a puzzle. The puzzle is to take an array (less than 10) and, starting at the beginning of the array, get to the one value that’s set to zero. The rules are that you can only move the number of spaces defined by the integer you are currently on. You can move in either direction given that it’s within the boundaries of the array.

**2. Planning the Solution**

**2.1 Solution Strategy**

The most critical part of this assignment is determining the means of traversing the array. I found it helpful to draw each node from the example out on paper and construct a graph linking each node. By doing this we can see that we have a directed graph and the path we are allowed to traverse with the given rules is essentially a tree graph with some cycles. This lends itself well to depth first search or breadth first search. Since the requirement is to use recursion however, we need to opt for DFS.

With that determined the next step is the framework of the program. We’ll need to create a vertex Model class so that we can store both the value of the node as well as it’s adjacency list. Additionally, we can store a Boolean value for if it’s been visited or not. Next we need to create a Graph Navigator class for creating the graph running depth first search and returning the traversal path. Lastly, we’ll need a basic JavaFX application to run in the Main class with animations.

**)2.2 Goal Decomposition**

*Sub-goal 1*   Create vertex model class

*Sub-goal 2* Create Graph Navigator class

*Sub-goal 3*   Create Main class with javaFX application and animations.

**2.3** **Resources**

JavaFX

**3 Source Code**

Source code is in a zip file.

**3 Program and Module Description**

Vertex

Model for a vertex data object. It has a field for the Value of the vertex, adjacency list (Edges), and isVisited.

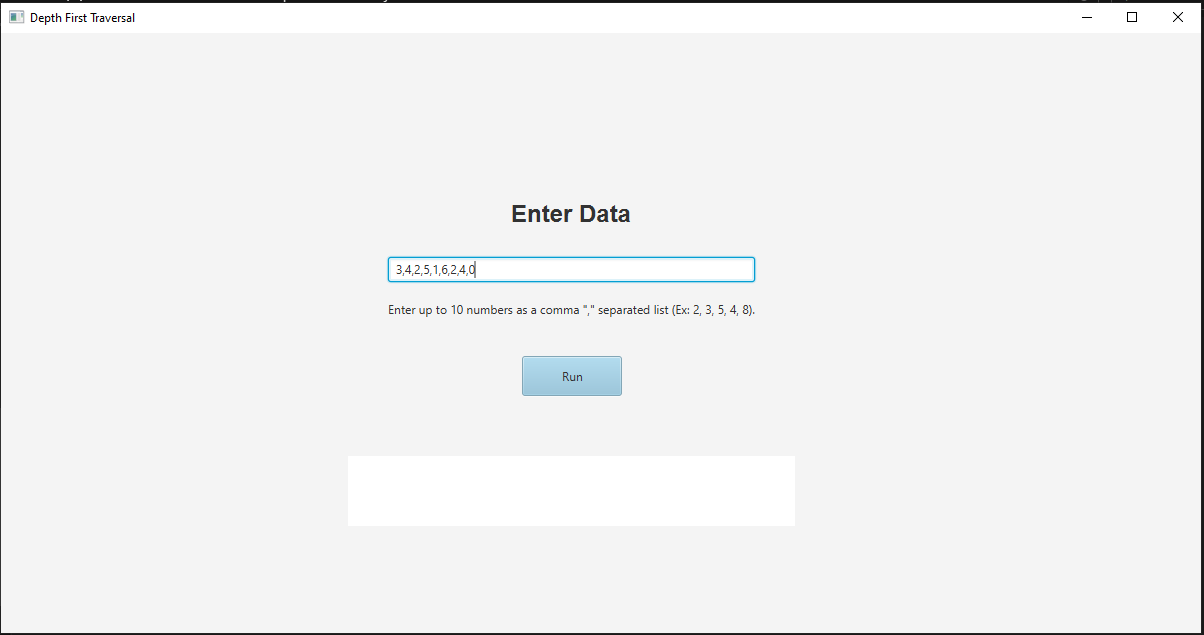
GraphNavigator

Contains functions for building a Graph from an ArrayList, traversing a graph with DFS, and returning an array of the traversal that stops when zero is reached or when all nodes have been visited (if they can be visited).

Main

Creates a JavaFX application with a root pane and 2 sub panes. It also adds nodes for data entry and a button. The button setOnAction triggers an event that collects the input data, feeds it into the GraphNavigator class, and receives the traversalPath array. The userArray is then used to create a visual depiction of the array from rectangle shapes, circles, and text. Lastly, the traversalPath array indicates when to fade the circles in and out using the keyframe and the timeline class.

**6. Input/Output**

****