Navid Hariri

David Quines

Computer Science 282

May 20, 2019

Status

# Status:

This project currently compiles and runs. Upon start the program it will read the text.txt file, create the graph, and then will displays a menu. When one is pressed the program displays weather the graph is connected or disconnected. When two is pressed it shows the edges and nodes needed for a minimum spanning tree. When pressing three it asks for a node to start from and then it gives the shortest path and cost to each node. When four is pressed the program ends. If any other numbers or letters are pressed the program gives an error and continues with the program, and the errors are caught.

# Development Process:

We first created the project with an adjacency list, and finished the driver, connected, and we did most of Kruskals algorithm, when we tried to do the shortest path using Dijkstra we had more difficulties and Kruskals algorithm was not fully implemented, and we decided to convert the whole graph to a matrix representation. We were able to finish Dijkstra shortest path algorithm, the minimum spanning tree, however, the connected method was a bit troublesome and we did try a few different methods until we noticed they all worked, and we used one of them for our driver. The program is heavily commented for readability and we tried making it self-documented.

# Difficulties:

The main difficulty that we had was working with the adjacency list representation and using it to do Kruskals algorithm and shortest path algorithm. We had lots of difficulties using that implementation to do the algorithms and understand how it works, and it was a long process to convert it to adjacency matrix, but we were able to finish it easier, which was worth it. We also originally had a vertex class that we ended up implementing inside the graph instead.