Jesus Otteson

Data Structures & Algorithms

Final CODE

07/31/2018

**Abstract**

This White Paper is to go over the specifications and details on how to create a program that will show a list of specific cities and their connections. This program will also displace the shortest path from any two cities and have a function to exit.

**Programs Design**

This program will have a GUI in its own class that will allow the user to Select from a drop down whether they would like to display the connecting cities available, the shortest path from any two cities and an option to exit the application. This program will use a minimum spanning tree that will take a selected city and compare it against any other city in this program. The minimum spanning tree will find the best route for the user to take by finding the path with the shortest distance. We will want to create another class for an edge weighted graph and use Prims Algorithm for the minimum spanning tree as this algorithm will find all routes from one point and find the best route to take by comparing all the closest cities not yet already connected when there is a choice to choose between two. This program should run with a Big O time of O(E Log V) making this algorithm the most efficient when deciding between using Prim, Dijkstra, or Kruskal’s algorithms. Dijkstra’s Big O time is 0(V^2) which wouldn’t make it the most effective, however, I believe Prim’s algorithm would be more effective than Kruskal’s algorithm as it would find the best path from the starting point rather than Kruskal’s algorithm that may work more effectively when plotting a trip with multiple stops as it would check routes by distance of all cities.