Ryan Madden

Assignment 6

12/2/2016

Traveling Sales Person and Stacks

Programing assignment six required a reimplementation of Lab Five's traveling sales person problem. We were tasked with finding the "cheapest path" from the starting city by traveling to each other city once and only once. The cities and cost were represented by an adjacency matrix with the travel cost to and from each city given. To solve this problem originally we had used a recursive depth first search algorithm to test the cost of every possible path using an array list to store and calculate the cost of each given path. For this assignment we implemented Dijkstra's algorithm to find the shortest path using a stack as a container for the current city and an integer array to denote if a city had been visited. The algorithm would start at point 0, and traverse to the closest point that had not been visited until all points were traversed. The path of the cities as the algorithm traversed them.

The runtimes for each N (number of cities) is shown in the table below. Using the recursive depth first search algorithm for N = 16 took several hours while N = 19 was left running overnight and still had not completed. Because of this N = 29 was never attempted. Our second algorithm at its longest, N = 29 completed quicker than N = 12 from the original method.

N = Number of Points	Recursive Depth First Search Array List	Dijkstra's with Stack
12	Seconds 0.424518	Seconds 4.18986E-4
13	Seconds 1.4368451	Seconds 3.74613E-4
14	Seconds 9.492309	Seconds 5.16693E-4
15	Seconds 17.653261	Seconds 4.46293E-4
16		seconds 4.71893E-4
19		Seconds 5.04319E-4
29		Seconds 0.001074346