C950 Task-2 WGUPS Algorithm Overview

Task-2: Implementation phase of the WGUPS Routing Program

Justin Ortiz

ID #010164085

WGU Email: jort291@wgu.edu

10/10/2023

C950 Data Structures and Algorithms II

F1. Justify the package delivery algorithm used in the solution as written in the original program by doing the following: One strength of the chosen algorithm is that it compartmentalizes tasks in order to complete. It does this by calling another function that finds the minimum distance between two addresses. This improves the ease of managing the code because potential problems can be better located if they are compartmentalized. The second strength of the algorithm is the fact that it is a greedy algorithm. This greedy algorithm ensures that it always finds the nearest address without compromise.

F3. Two algorithms that could be used to meet all requirements are the Djikstra's algorithm and Bellman Ford's algorithm.

F3a. Djikstra's algorithm differs from my program algorithm by utilizing a graphical approach to finding the shortest path. The Bellman Ford's algorithm differs from my program by its comparison strategy. This algorithm searches for the shortest point that it can get to in one step, then two steps, and so on.

G. If I could do this project again, I would choose a named algorithm, manage the delayed packages better, and utilize a better time keeping system. I would choose a named algorithm because these algorithms, such as the Djikstra's algorithm are well-established and proven to be efficient. I would manage the delayed packages better by creating the capability to add them to the unvisited list at whatever time I needed to. I would utilize a better time keeping system by creating a way to search up the trucks at whatever time I needed to instead of only updating the time whenever the truck makes another delivery.

H1. Two other data structures that could meet the same requirements in this scenario are binary search trees and linked lists.

H1a. Both binary search trees and linked lists differ from a hash table by the way they store their data. A hash table stores its data by hashing a key into a bucket. A binary search tree differs from this by storing its data in parent-child structures. Each node is able to have two children which represent less than or greater than. A linked list stores its data by using nodes which have a previous and next node pointer.