Joseph Bonhomme

C195

Task 2

Task 2 Write-Up

Here is the completed code with total miles:

A screenshot of a computer

Description automatically generated

Continues from package 26 down

A screenshot of a computer

Description automatically generated

Here are the package statuses at 8:35am:

A screenshot of a computer

Description automatically generated

Continues from package ID 17

A screenshot of a computer

Description automatically generated

The following image shows the package statuses from 10:00 am

A screenshot of a computer

Description automatically generated

From package 25 and down

A screenshot of a computer

Description automatically generated

The following image shows the package statuses at 1pm (13:00 military time)

A screenshot of a computer

Description automatically generated

This is from package ID 24 down

A screenshot of a computer

Description automatically generated

F1. There are several strengths to the algorithm I used which I will discuss. One of the strengths is that this is a simple algorithm to implement. There is no need for additional data structures such as a graph to implement this, it only requires an array and the ability to find the shortest distance between two addresses. Another strength is that the nearest neighbor algorithm can be applied to wide range of problems like regression. It is extremely useful since it is such an easy algorithm to implement.

F2. I have verified that all requirements have been met within the provided scenario. All packages are delivered on time and all packages are assigned the right truck. Package 9’s address is updated at 10:20 so my solution was to make truck 3 leave at 10:20 since truck 1 returns before then(You can run the deliver function and see when each truck returns from their deliveries). The algorithm consistently finds the shortest distance and travels to that location until all packages on the truck have been delivered.

F3.Two other algorithms that I could’ve potentially used for this assignment are Dijkstra’s algorithm and the A\* search algorithm. Dijkstra’s as well as the A\* search algorithm are different from nearest neighbor because they rely on using a graph for their implementation. Dijkstra’s would work because it calculates the shortest possible path from the starting location to the end location. A\* Search is like Dijkstra’s, but it uses a weighted graph instead of an unweighted graph. It focuses on finding the optimal path from one location to another instead of nearest neighbor where the optimal path is not calculated. Since these find the optimal distances it would work for this given problem, but they both would be much more difficult to implement.

G. If I were to do this project differently, I would find a way to optimize the time complexity of my implementation. I would use less for loops and investigate other built in data structures and methods that I could use to further optimize the time complexity of my algorithm.

H. The hash table I used satisfies the data structure requirement since it performs all necessary functions outlined in the task overview. Another data structure that I could have used would be a Doubly Linked List where each node would contain package data. It works like a list, but it is much more complicated for this assignment. It would work however it would be a lot of code to implement correctly. A linked List is different because it doesn’t rely on key value pairs in a bucket like a hash table but instead each separate Node has its own data assigned to it. Another data structure I could have used would be a hash map where each key could be the package ID and the value would be the corresponding package object. A HashMap is different from a hash table because a HashMap uses a dictionary for its implementation while a hash table uses an array of buckets.

I. Sources: GeeksforGeeks. (2023, March 8). *A\* search algorithm*. GeeksforGeeks. https://www.geeksforgeeks.org/a-search-algorithm/

GeeksforGeeks. (2023c, November 23). *Find shortest paths from source to all vertices using Dijkstra’s algorithm*. GeeksforGeeks. https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/