C950 Task-2 WGUPS Write-Up

(Task-2: The implementation phase of the WGUPS Routing Program).

(Zip your source code and upload it with this file)

Kevin Salazar

ID # 010303855

WGU Email: ksala83@wgu.edu

December 27th, 2023

C950 Data Structures and Algorithms II

# A. Hash Table

Hash Table Screenshot goes here

A computer screen shot of a program code

Description automatically generated

# B. Look-Up Functions

Look-up function screenshot goes here

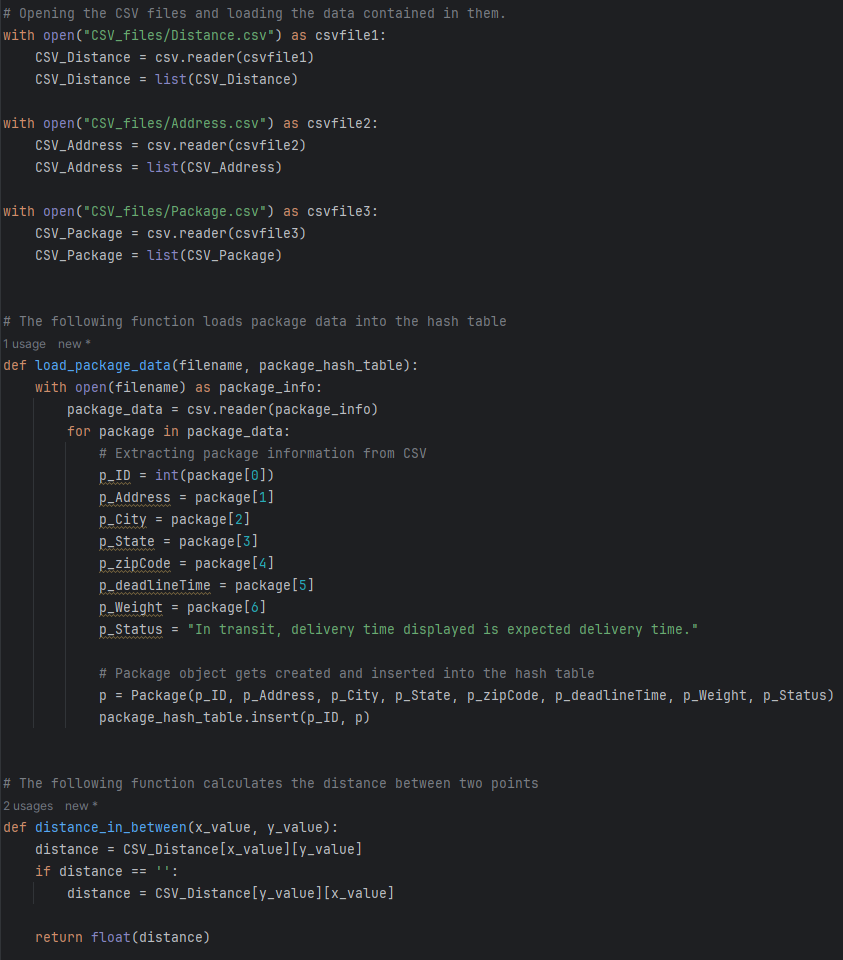
A computer screen shot of a program code

Description automatically generated

# C. Original Code

Major code blocks screenshots go here showing implementation.

**From main.py, showing the code necessary to load the data from the csv files. Also, a function to load the package data into a package hash table. Ending with a function that calculates distance between two points.**



**Function that gets the address information from the csv files.**

A screen shot of a computer

Description automatically generated

# C1. Identification Information

main.py screenshot goes here showing Student ID

A screen shot of a computer

Description automatically generated

# C2. Process and Flow Comments

Some code blocks screenshots go here showing comments.

**Creation of truck objects to place the specific packages that belong to each one of them. Also, creating a hash map for package storage and loading the package information from the csv file into the hash table.**

A screenshot of a computer

Description automatically generated

**Function that simulates the package delivery, using a truck object. Creating an empty list of packages that have not been delivered yet, adding all packages that belong to that truck in that same list. After that, a while loop that delivers the packages, until the truck list with packages is empty.**

A computer screen shot of a program

Description automatically generated

**Calling previous mentioned function in all three trucks, for execution of delivery process, also placing the specific departure time for the third truck.**

A screen shot of a computer

Description automatically generated

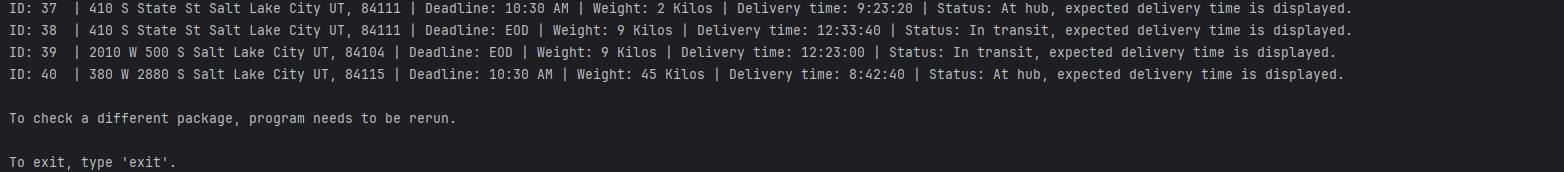
**The update method that fixes the issue with package 9, not having its address correctly updated before/after 10:20AM**

**A screen shot of a computer program

Description automatically generated**

# D. Interface

Interface screenshot goes here **8:30AM. The user can view the total mileage for each truck as well as the total mileage for the route, along with all the packages.**A screen shot of a computer

Description automatically generated 

# D1. First Status Check

Provide screenshots to show the status of all packages loaded onto each truck at a time between 8:35 a.m. and 9:25 a.m. **Checking the status of packages at 9:00AM.**

A screen shot of a computer screen

Description automatically generated

# D2. Second Status Check

Provide screenshots to show the status of all packages loaded onto each truck at a time between 9:35 a.m. and 10:25 a.m. **Checking the status at 10:23AM.**

**A screen shot of a computer screen

Description automatically generated**

# D3. Third Status Check

Provide screenshots to show the status of all packages loaded onto each truck at a time between 12:03 p.m. and 1:12 p.m. **Checking status at 12:59PM**

A screen shot of a computer screen

Description automatically generated

# E. Screenshot of Code Execution

Provide screenshots showing successful completion of the code that includes the total mileage traveled by all trucks. **Code and successful completion of code.**

A computer screen shot of a program

Description automatically generatedA screenshot of a computer screen

Description automatically generated

# F1. Strengths of the Chosen Algorithm

The nearest neighbor algorithm implemented in the package delivery process shows efficiency by presenting a solution where all 40 packages get delivered on time while meeting each package’s requirements and keeping the combined total distance traveled under 140 miles.

It also provides a great time complexity, because of the use of the hash map to store all the package information, lookup, which benefits the goal of delivering on time.

# F2. Verification of Algorithm

The nearest neighbor algorithm (which uses proximity to make the decision on which package is next on the delivery list, using its data) completes all requirements in the described scenario because:

The packages along with their constraints (like those that were not getting to the hub until 9:05, they could not leave for delivery until after 9:05, the packages that were supposed to be delivered in a specific truck were delivered as requested, with their specific truck) are delivered on time.

The route is less than 140 miles.

The supervisor can see, at different times, the progress of each truck and its packages by checking its status, on a list or by individual package.

If a package is already delivered at the report time, it is displayed. And if it has not been delivered yet, an estimated delivery time is displayed.

# F3. Other Possible Algorithms

Geocoding Algorithm: Used to convert addresses into geographic coordinates (called geocoding) and vice versa (reverse geocoding), can be crucial for accurately pinpointing delivery locations on maps. Helping a lot for rural areas where delivery points need to be specifically placed as maps are not updated as often as maps in urban areas.

Dijkstra's Algorithm: Used for finding the shortest path between nodes in a graph. It can optimize route planning by considering distances or travel times between various delivery locations. But in this scenario, it would be way more specific, involving many other parameters that would affect the travel times, like weather conditions, traffic conditions, construction near the routes, etc.

# F3a. Algorithm Differences

The difference between the nearest neighor algorithm used in the project and the Geocoding Algorithm, is that the address information was not taken from the geo coordinates of the delivery points. Also, compared to Geocoding it can be less accurate for delivery points found in rural areas. A driver can be driving to a “delivery point” but it could be the entrance point of a street where the address is located, or also not just be accurate to what the address seems to appear in online map systems.

The difference between the algorithm used in the project and the Dijkstra’s algorithm is that, the distance was taken based on (x, y) values only, whereas if Dijkstra was used, you would have to be more specific to develop super accurate delivery times, involving all types of conditions that could be found in the delivery process. If it was snowing, delivery time would be longer, as driving conditions change and driver is needed to drive with more caution, considering a system in case a package gets delayed and needs to be delivered the next day.

# G. Different Approach

A different approach on this project would be by optimizing the project by creating a priority queue based on the distance between the truck's current location and the package addresses, instead of iterating through all undelivered packages each time. This queue would involve priority values.

Also, by using Dijkstra's algorithm to extract the next package with the shortest distance from the priority queue.

# H. Verification of Data Structure

The hash table was the data structure used and it meets all the requirements for the scenario, because:

The package ID acts as the perfect key match for the hash table.

All package components (address, city state, zip, weight, status, notes, and deadline) are stored as values for the matching key.

The hash table makes package data easy to retrieve and manipulate for the program needs (searching by package ID, removing a package, adding a new package, loading certain packages into specific trucks, etc.)

# H1. Other Data Structures

Priority queue, by adding a priority value system where the deliveries are guided by these mentioned values, it would avoid having to iterating processes, over and over. Making it more effective for a larger number of packages, and also to adjust priority in case of any changes during the delivery (if the customer preferred to pick up the package at a near pick-up location, then the original priority value would change to the situation).

In Python, by using Dictionaries. dictionaries are built-in data structures that store key-pair values. Storing undelivered packages in the dictionary with package IDs as keys. Dictionaries use hashing mechanisms, making it similar to a Hash Map (which are different and more generic).

As the dictionary data structure is specific for Python, but very similar to Hash Map, a third Data structure will be mentioned.

Graphs could be implemented in this project, matching vertices/nodes with addresses and edges for easier route planning and creation. Implementing the adjacency matrix, and algorithms for better organization.

# H1a. Data Structure Differences

The hash map used in the project was efficient because it met all requirements, but if delivery points were businesses that required a delivery to be made before the closing time of the business, priority queues could optimize the delivery process based on package deadlines, and not just the distance between the delivery points.

Dictionaries are very similar to Hash maps, so the difference would be minimal, it gives faster insertion, deletion, and retrieval of packages. But with hashing methods that were already implemented.

A Graph would make a difference in the placement of delivery points, as you would be able to introduce an Adjacent matrix for better management. “Graph algorithms have a non-linear data structure of edges and nodes. Breadth-first search algorithm may be used to discover the locations nearest to a specific origin point.” (Memgraph, 2022). Would be a great idea if there were a lot of delivery points that made Hash Map more inefficient. Also, algorithms involving graphing can be applied, optimizing the system even more.

# I. Sources

Applications of the 20 Most Popular Graph Algorithms (March 11, 2022) by Memgraph

Retrieved December 26th, 2023, from

https://memgraph.com/blog/graph-algorithms-applications

# J. Professional Communication