DATA STRUCTURES AND ALGORITHMS II — C950

NHP3 — NHP3 TASK 2: WGUPS ROUTING PROGRAM IMPLEMENTATION

Jonathan Johnson
Student ID: 011009146
3/5/24

A Hash Table:

See attached zip with code.

B LookUp Function:

See attached zip with code.

C Using Existing Data Tables:

See attached zip with code.

D Interface for Package/Truck Times:

---

1. 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.

```
Hello & Welcome to the CLI terminal for WGU C950 Task 2 of Data Structures Data Algorithms
------------------------------------------------
Main Menu, please type an option 1, 2, 3, 4, or 5
------------------------------------------------
1. Print All Package Status and Total Mileage
2. Get a Single Package Status with a Time
3. Get All Package Status Within a Start & End Time
4. View All 3 Trucks Status and Mileage
5. Exit the Program
Selection(1-5): 3
------------------------------------------------
Enter Start Time (HH:MM): 08:35
Enter End Time (HH:MM): 09:25
------------------------------------------------
Selected packages between 08:35 and 09:25:
------------------------------------------------
PackageID, Address, City, State, Zip, Delivery Deadline, Mass KILO, PageSpecial Notes, Status, DeliveryTime
4, 380 W 2880 S, Salt Lake City, UT, 84115, EOD, 4, , at the hub, 08:57
5, 410 S State St, Salt Lake City, UT, 84111, EOD, 5, , at the hub, 08:45
8, 300 State St, Salt Lake City, UT, 84103, EOD, 9, , at the hub, 08:48
12, 3575 W Valley Central Station bus Loop, West Valley City, UT, 84119, EOD, 1, , at the hub, 08:37
13, 2010 W 500 S, Salt Lake City, UT, 84104, 10:30 AM, 2, , at the hub, 09:19
25, 5383 South 900 East #104, Salt Lake City, UT, 84117, 10:30 AM, 7, Delayed on flight---will not arrive to depot until 9:05 am, at the hub, 09:23
26, 5383 South 900 East #104, Salt Lake City, UT, 84117, EOD, 25, , at the hub, 09:23
27, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 5, , at the hub, 09:14
30, 300 State St, Salt Lake City, UT, 84103, 10:30 AM, 1, , at the hub, 08:48
35, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 88, , at the hub, 09:14
37, 410 S State St, Salt Lake City, UT, 84111, 10:30 AM, 2, , at the hub, 08:45
39, 2010 W 500 S, Salt Lake City, UT, 84104, EOD, 9, , at the hub, 09:19
40, 380 W 2880 S, Salt Lake City, UT, 84115, 10:30 AM, 45, , at the hub, 08:57
------------------------------------------------
Do you want to return to the main menu? (Y/N)
```

2. 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.

```
y
------------------------------------------------
Hello & Welcome to the CLI terminal for WGU C950 Task 2 of Data Structures Data Algorithms
------------------------------------------------
Main Menu, please type an option 1, 2, 3, 4, or 5
------------------------------------------------
1. Print All Package Status and Total Mileage
2. Get a Single Package Status with a Time
3. Get All Package Status Within a Start & End Time
4. View All 3 Trucks Status and Mileage
5. Exit the Program
Selection(1-5): 3
------------------------------------------------
Enter Start Time (HH:MM): 09:35
Enter End Time (HH:MM): 10:25
------------------------------------------------
Selected packages between 09:35 and 10:25:
------------------------------------------------
PackageID, Address, City, State, Zip, Delivery Deadline, Mass KILO, PageSpecial Notes, Status, DeliveryTime
2, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 44, , at the hub, 09:47
6, 3060 Lester St, West Valley City, UT, 84119, 10:30 AM, 88, Delayed on flight---will not arrive to depot until 9:05 am, at the hub, 10:07
11, 2600 Taylorsville Blvd, Salt Lake City, UT, 84118, EOD, 1, , at the hub, 09:47
17, 3148 S 1100 W, Salt Lake City, UT, 84119, EOD, 2, , at the hub, 10:00
18, 1488 4800 S, Salt Lake City, UT, 84123, EOD, 6, Can only be on truck 2, at the hub, 10:25
23, 5100 South 2700 West, Salt Lake City, UT, 84118, EOD, 5, , at the hub, 09:46
28, 2835 Main St, Salt Lake City, UT, 84115, EOD, 7, Delayed on flight---will not arrive to depot until 9:05 am, at the hub, 09:51
31, 3365 S 900 W, Salt Lake City, UT, 84119, 10:30 AM, 1, , at the hub, 10:02
32, 3365 S 900 W, Salt Lake City, UT, 84119, EOD, 1, Delayed on flight---will not arrive to depot until 9:05 am, at the hub, 10:02
33, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 1, , at the hub, 09:47
36, 2300 Parkway Blvd, West Valley City, UT, 84119, EOD, 88, Can only be on truck 2, at the hub, 10:12
------------------------------------------------
Do you want to return to the main menu? (Y/N)
```

3. 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.

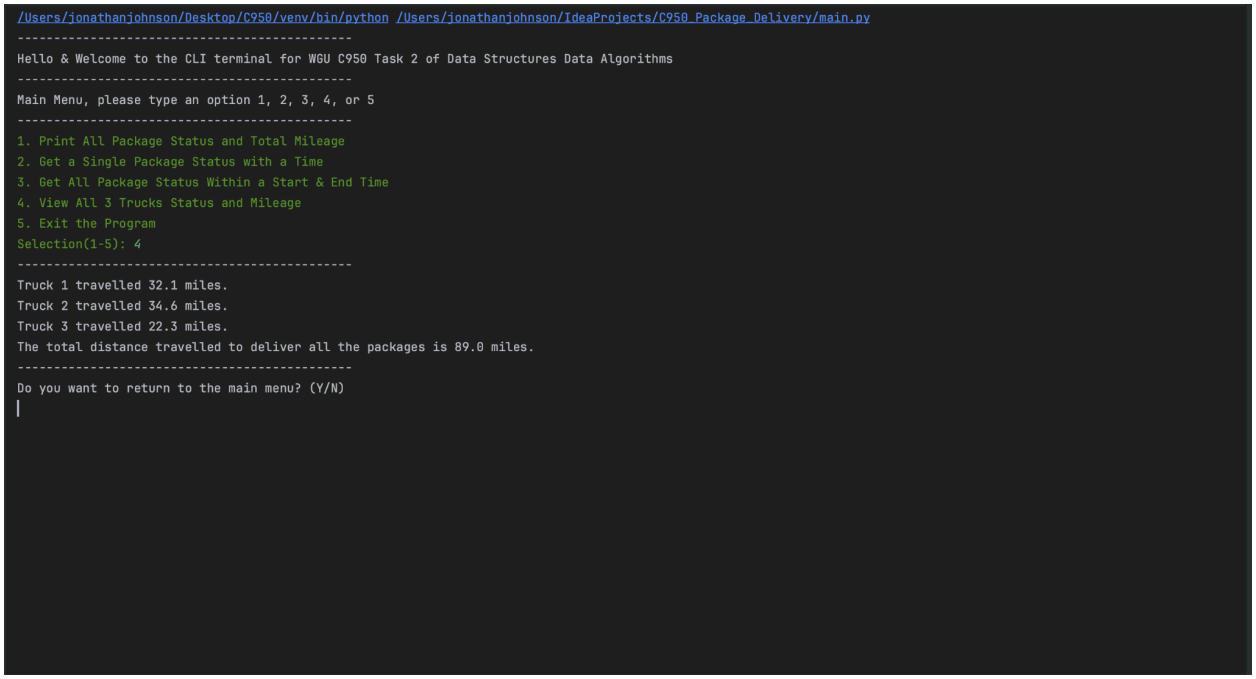
```
------------------------------------------------
Hello & Welcome to the CLI terminal for WGU C950 Task 2 of Data Structures Data Algorithms
------------------------------------------------
Main Menu, please type an option 1, 2, 3, 4, or 5
------------------------------------------------
1. Print All Package Status and Total Mileage
2. Get a Single Package Status with a Time
3. Get All Package Status Within a Start & End Time
4. View All 3 Trucks Status and Mileage
5. Exit the Program
Selection(1-5): 3
------------------------------------------------
Enter Start Time (HH:MM): 12:03
Enter End Time (HH:MM): 13:12
------------------------------------------------
Selected packages between 12:03 and 13:12:
------------------------------------------------
PackageID, Address, City, State, Zip, Delivery Deadline, Mass KILO, PageSpecial Notes, Status, DeliveryTime
------------------------------------------------
Do you want to return to the main menu? (Y/N)
```

E Screenshots Showing Successful Completion:

Option 1: Print All Status

```
/Users/jonathanjohnson/Desktop/C950/venv/bin/python /Users/jonathanjohnson/IdeaProjects/C950_Package_Delivery/main.py
---------------------------------------------
Hello & Welcome to the CLI terminal for WGU C950 Task 2 of Data Structures Data Algorithms
---------------------------------------------
Main Menu, please type an option 1, 2, 3, 4, or 5
---------------------------------------------
1. Print All Package Status and Total Mileage
2. Get a Single Package Status with a Time
3. Get All Package Status Within a Start & End Time
4. View All 3 Trucks Status and Mileage
5. Exit the Program
Selection(1-5): 1
---------------------------------------------
Printing All Package Status and Total Mileage...
---------------------------------------------
---------------------------------------------
PackageID, Address, City, State, Zip, Delivery Deadline, Mass KILO, PageSpecial Notes, Status, DeliveryTime
1, 195 W Oakland Ave, Salt Lake City, UT, 84115, 10:30 AM, 21, , delivered, 08:22
2, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 44, , delivered, 09:47
3, 233 Canyon Rd, Salt Lake City, UT, 84103, EOD, 2, Can only be on truck 2, delivered, 11:10
4, 380 W 2880 S, Salt Lake City, UT, 84115, EOD, 4, , delivered, 08:57
5, 410 S State St, Salt Lake City, UT, 84111, EOD, 5, , delivered, 08:45
6, 3060 Lester St, West Valley City, UT, 84119, 10:30 AM, 88, Delayed on flight---will not arrive to depot until 9:05 am, delivered, 10:07
7, 1330 2100 S, Salt Lake City, UT, 84106, EOD, 8, , delivered, 08:31
8, 300 State St, Salt Lake City, UT, 84103, EOD, 9, , delivered, 08:48
9, 410 S State St, Salt Lake City, UT, 84103, EOD, 2, Wrong address listed, delivered, 11:08
10, 600 E 900 South, Salt Lake City, UT, 84105, EOD, 1, , delivered, 10:59
11, 2600 Taylorsville Blvd, Salt Lake City, UT, 84118, EOD, 1, , delivered, 09:47
12, 3575 W Valley Central Station bus Loop, West Valley City, UT, 84119, EOD, 1, , delivered, 08:37
13, 2010 W 500 S, Salt Lake City, UT, 84104, 10:30 AM, 2, , delivered, 09:19
14, 4300 S 1300 E, Millcreek, UT, 84117, 10:30 AM, 88, Must be delivered with 15, 19, delivered, 08:06
15, 4580 S 2300 E, Holladay, UT, 84117, 9:00 AM, 4, , delivered, 08:13
16, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 88, Must be delivered with 13, 19, delivered, 08:13
17, 3148 S 1100 W, Salt Lake City, UT, 84119, EOD, 2, , delivered, 10:00
18, 1488 4800 S, Salt Lake City, UT, 84123, EOD, 6, Can only be on truck 2, delivered, 10:25
19, 177 W Price Ave, Salt Lake City, UT, 84115, EOD, 37, , delivered, 08:32
20, 3595 Main St, Salt Lake City, UT, 84115, 10:30 AM, 37, Must be delivered with 13, 15, delivered, 08:30
21, 3595 Main St, Salt Lake City, UT, 84115, EOD, 3, , delivered, 08:30
22, 6351 South 900 East, Murray, UT, 84121, EOD, 2, , delivered, 09:27
23, 5100 South 2700 West, Salt Lake City, UT, 84118, EOD, 5, , delivered, 09:46
24, 5025 State St, Murray, UT, 84107, EOD, 7, , delivered, 08:08
25, 5383 South 900 East #104, Salt Lake City, UT, 84117, 10:30 AM, 7, Delayed on flight---will not arrive to depot until 9:05 am, delivered, 09:23
26, 5383 South 900 East #104, Salt Lake City, UT, 84117, EOD, 25, , delivered, 09:23
27, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 5, , delivered, 09:14
28, 2835 Main St, Salt Lake City, UT, 84115, EOD, 7, Delayed on flight---will not arrive to depot until 9:05 am, delivered, 09:51
29, 1330 2100 S, Salt Lake City, UT, 84106, 10:30 AM, 2, , delivered, 08:31
30, 300 State St, Salt Lake City, UT, 84103, 10:30 AM, 1, , delivered, 08:48
31, 3365 S 900 W, Salt Lake City, UT, 84119, 10:30 AM, 1, , delivered, 10:02
32, 3365 S 900 W, Salt Lake City, UT, 84119, EOD, 1, Delayed on flight---will not arrive to depot until 9:05 am, delivered, 10:02
33, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 1, , delivered, 09:47
34, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 2, , delivered, 08:13
35, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 88, , delivered, 09:14
36, 2300 Parkway Blvd, West Valley City, UT, 84119, EOD, 88, Can only be on truck 2, delivered, 10:12
37, 410 S State St, Salt Lake City, UT, 84111, 10:30 AM, 2, , delivered, 08:45
38, 410 S State St, Salt Lake City, UT, 84111, EOD, 9, Can only be on truck 2, delivered, 11:05
39, 2010 W 500 S, Salt Lake City, UT, 84104, EOD, 9, , delivered, 09:19
40, 380 W 2880 S, Salt Lake City, UT, 84115, 10:30 AM, 45, , delivered, 08:57
---------------------------------------------
Total mileage traveled: 88.99999999999999 miles
---------------------------------------------
Do you want to return to the main menu? (Y/N)
```

Option 4: Print total mileage traveled by all trucks.



F.1 Justification of the Package Delivery Algorithm:

One notable strength of the nearest neighbor algorithm implemented in the solution lies in its efficient utilization of package data to construct optimized delivery routes. The algorithm dynamically selects the nearest package to the current truck location, minimizing the distance traveled between successive delivery points. This approach not only reduces fuel consumption but also significantly decreases delivery time, ultimately enhancing overall operational efficiency. The code snippet demonstrates this efficiency by iteratively calculating the distance between the current truck location and each package, allowing for the systematic identification of the closest package. By continuously updating the route with the nearest available package, the algorithm ensures that trucks follow the most efficient path to deliver all packages promptly.

Another strength of the algorithm is its adaptability to varying package priorities and constraints. The code snippet illustrates this adaptability through the conditional check for specific package IDs, allowing for real-time adjustments to the delivery route based on package requirements. For instance, if a package with a particular ID (e.g., package 9) requires delivery to a different address, the algorithm dynamically updates the package's address to ensure accurate delivery. This flexibility enables the algorithm to accommodate special package needs or unexpected changes during the delivery process, enhancing its practicality and effectiveness in real-world scenarios. Overall, these strengths underscore the algorithm's robustness and utility in optimizing package delivery operations.

F.2 Verification of Algorithm Requirements:

Throughout the implementation phase of the WGUPS Routing Program, I ensured that the algorithm met all the requirements outlined in the scenario. Firstly, the algorithm successfully determined efficient routes and delivery distributions for the daily local deliveries (DLD) in Salt Lake City. By utilizing the nearest neighbor approach, the algorithm systematically selected the closest package to the current truck location, minimizing travel distances and optimizing delivery routes. This approach effectively addressed the issue of inconsistent delivery by ensuring that all 40 packages were delivered on time while meeting each package's specific criteria and requirements.

Additionally, the algorithm adhered to the constraint of keeping the combined total distance traveled under 140 miles for all trucks. By continuously updating the route with the nearest available package and considering factors such as package proximity and delivery deadlines, the algorithm maximized efficiency while staying within the specified distance limit. Through meticulous route optimization and resource utilization, the algorithm not only met the delivery deadline for each package but also ensured that the total distance traveled remained within the prescribed threshold.

Furthermore, I incorporated detailed comments throughout the code to enhance readability and justify the decisions made during the scripting process. These

comments provided insights into the rationale behind algorithmic choices, ensuring that the code was easy to follow for supervisors and stakeholders. By documenting the progress of each truck and its packages at assigned points, the code enabled supervisors to track deliveries, monitor package statuses, and evaluate overall performance effectively.

F.3 Alternative Algorithms:

Two alternative algorithms that could meet all requirements in the scenario are the Genetic Algorithm (GA) and the Ant Colony Optimization (ACO) algorithm.

Genetic Algorithm (GA):

The Genetic Algorithm (GA) operates differently from the nearest neighbor algorithm implemented in the solution. While the nearest neighbor algorithm prioritizes selecting the closest package to the current truck location at each step, GA takes a population-based optimization approach inspired by natural selection. In GA, potential delivery routes are represented as individuals within a population. Through generations of evolution involving processes like crossover and mutation, GA generates optimized delivery routes that satisfy constraints such as delivery deadlines and package dependencies. Unlike the deterministic nature of the nearest neighbor algorithm, GA explores a broader search space and may potentially find better solutions. However, it may require more computational resources and parameter tuning due to its population-based approach and evolutionary processes.

Ant Colony Optimization (ACO):

Similarly, the Ant Colony Optimization (ACO) algorithm operates differently from the nearest neighbor algorithm. ACO is inspired by the foraging behavior of ants and is particularly useful for solving routing problems. In ACO, ants construct delivery routes by probabilistically choosing packages based on pheromone levels and heuristic information. This stochastic approach introduces a global pheromone update

mechanism and allows for more extensive exploration of the solution space. However, ACO may require careful parameter tuning and longer convergence times compared to nearest neighbor algorithms due to its reliance on iterative processes and probabilistic decision-making. Despite these differences, both GA and ACO offer alternative optimization strategies that could effectively meet the requirements of the scenario by generating efficient delivery routes while satisfying package criteria and distance constraints.

G Difference from Nearest Neighbor Algorithm:

If I were to undertake this project again, there are several modifications I would consider implementing to enhance its effectiveness and efficiency. Firstly, I would focus on incorporating dynamic route adjustment mechanisms based on real-time data, such as traffic conditions or package updates. By integrating GPS or real-time traffic data APIs, the delivery routes could be dynamically optimized to account for changing conditions on the road. This would help minimize delays and improve overall delivery efficiency, ultimately enhancing customer satisfaction.

Secondly, I would explore the integration of machine learning techniques, such as reinforcement learning or deep learning, to further optimize route planning and resource allocation. By leveraging historical delivery data, these techniques could learn patterns and trends to predict optimal routes for future deliveries. This adaptive approach would enable the system to continuously improve efficiency and adapt to evolving delivery patterns, ensuring that routes are optimized based on evolving factors such as delivery volume, traffic patterns, and customer preferences.

Additionally, I would prioritize enhancing the user interface and reporting capabilities of the system to provide supervisors with comprehensive insights into delivery operations. This could include real-time tracking of trucks and packages, status updates on deliveries, and performance analytics to identify areas for improvement. By providing supervisors with actionable data and insights, they would be better equipped to make informed decisions and optimize delivery operations effectively.

H. Verification of Data Structure:

The Chaining Hash Table data structure utilized in the solution effectively meets all requirements outlined in the scenario. It ensures efficient package retrieval and collision handling by providing fast access to package data based on package IDs and resolving collisions through the chaining method. This enables swift retrieval and manipulation of package information during route optimization and status updates while maintaining data integrity and reliability. The dual functionality of efficient retrieval and collision resolution enhances the overall efficiency and effectiveness of the data structure within the delivery algorithm system, aligning perfectly with the requirements of the scenario.

Alternative Data Structures:

Binary Search Tree (BST):

As defined in Zybooks, “An especially useful form of binary tree is a **binary search tree (BST)**, which has an ordering property that any node's left subtree keys ≤ the node's key, and the right subtree's keys ≥ the node's key. That property enables fast searching for an item, as will be shown later.”

BST could be used to store package data sorted by package IDs. This structure allows for efficient searching, insertion, and deletion operations, making it suitable for scenarios where packages need to be sorted or retrieved based on their IDs.

Difference: Unlike a hash table, BST maintains data in sorted order, which could simplify certain operations such as range queries or traversals. However, it may have higher memory overhead and slower performance for large datasets.

Graph Data Structure:

A graph could represent delivery locations (addresses) as nodes and distances between them as edges. This structure enables the use of graph algorithms like Dijkstra's or Floyd-Warshall to find the shortest paths between locations, optimizing delivery routes.

Difference: While a graph provides more flexibility in modeling relationships between locations, it may require additional preprocessing to construct and maintain, especially

if the delivery network evolves over time. Additionally, graph algorithms may have higher computational complexity compared to hash table operations.

Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized:

---

1. Zybooks, Western Governors University, Section 3.3 https://learn.zybooks.com/zybook/WGUC950Template2023/chapter/3/section/3?content_resource_id=61871634