**NHP3 – NHP3 TASK 2**

**Ayomide Adedeji**

**Data Structures and Algorithms II – C950**

**Western Governors University**

**Student #007440467**

**F.  Justify the package delivery algorithm used in the solution as written in the original program by doing the following:**

**1.  Describe two or more strengths of the algorithm used in the solution.**

**Answer:**

* One strength of the algorithm used in this solution is its efficiency in package delivery optimization. The algorithm employs the nearest neighbor approach, which selects the next package for delivery based on its proximity to the current location of the truck. By choosing the nearest package iteratively, the algorithm minimizes the distance traveled by each truck, thus optimizing the overall delivery process.
* Another strength is the utilization of a hash table data structure to efficiently store and retrieve package information. Hash tables offer constant-time (O(1)) average case complexity for insertion, deletion, and lookup operations, making them suitable for quickly accessing package data during the delivery process. This enables fast retrieval of package details, such as addresses and delivery deadlines, which are essential for route planning and timely delivery execution.

**2.  Verify that the algorithm used in the solution meets all requirements in the scenario.**

**Answer:**

The algorithm used in the solution meets all requirements because each of the packages are delivered before their specified deadlines. The trucks all contained fewer than 16 packages and delivered all packages in 105.1 miles total. All the package’s specific criteria were also met such as the delivery time window, notes, and address.

**3.  Identify two other named algorithms that are different from the algorithm implemented in the solution and would meet all requirements in the scenario**.

Two other named algorithms that are different from the nearest neighbor algorithm implemented in the solution are:

1. **Dijkstra's Algorithm:** This algorithm is used to find the shortest path between nodes in a graph, which could be applied in this scenario to optimize the routes taken by the trucks.

2. **(A-star) Algorithm:** This is another popular pathfinding algorithm commonly used in scenarios like route planning. It incorporates heuristics to efficiently search for the shortest path between nodes.

1. **Describe how both algorithms identified in part F3 are different from the algorithm used in the solution.**

**Answer:** Both Dijkstra's algorithm and (A -star) algorithm differ from the nearest neighbor algorithm used in the solution in terms of their approach to finding paths and their considerations of distance or cost.

**G.  Describe what you would do differently, other than the two algorithms identified in part F3, if you did this project again, including details of the modifications that would be made.**

**Answer:**

If I were to do this project again, I would consider using optimized package loading by loading packages dynamically as they are needed. For example, load packages for each truck based on their route or load packages in batches to reduce memory usage. I would also work on the error handling and validation by strengthening error handling and input validation mechanisms to handle unexpected scenarios gracefully. For example, handle invalid user inputs, address mismatches, or missing package data more robustly to prevent program crashes.

**H.  Verify that the data structure used in the solution meets all requirements in the scenario.**

The Chaining Hash Table was the data structure used in this solution.

1. Insertion and Update: The `insert` method inserts a key-value pair into the hash table. It checks if the key already exists in the table, and if so, it updates the corresponding value. This functionality is crucial for updating package information as the delivery progresses. So, the code satisfies this requirement.

2. Lookup: The `lookup` method retrieves the value associated with a given key from the hash table. This functionality is necessary for accessing package information based on package ID or any other identifying key. Thus, this meets the requirement.

3. Removal: The `remove` method removes a key-value pair from the hash table based on the provided key. Although the scenario does not explicitly mention removing packages, having the ability to remove items is generally useful for maintaining data integrity.

A screenshot of a computer program

Description automatically generated

**1.  Identify two other data structures that could meet the same requirements in the scenario.**

**Answer:**

* **Graph-Based data structure**: A graph-based structure, such as an adjacency list or an adjacency matrix, could be used to represent the delivery locations and distances between them. Each node in the graph represents a delivery location, and edges represent the distances between locations. This structure allows for efficient traversal and pathfinding algorithms like Dijkstra's algorithm or A-star search to find optimal routes for package delivery.
* **Priority Queue data structure**: A priority queue data structure could be used to prioritize package deliveries based on their deadlines, delivery requirements, and other criteria. Packages with earlier deadlines or higher priority could be dequeued and delivered first, ensuring timely delivery while optimizing route efficiency.

1. **Describe how each data structure identified in H1 is different from the data structure used in the solution.**

**Answer:**

**Graph-Based data structure:** Unlike a chaining hash table, which is primarily used for fast key-value lookups, a graph-based structure focuses on representing relationships between entities. While a chaining hash table could be used to store package information and retrieve it quickly, it doesn't inherently capture the spatial relationships between delivery locations or facilitate route optimization.

**Priority Queue data structure:** While both priority queues and chaining hash tables can prioritize and manage data based on certain criteria, they serve different purposes. A priority queue is specifically designed for efficient retrieval and removal of elements based on priority, making it suitable for managing tasks or events with different levels of urgency. In contrast, a chaining hash table is more suited for associative arrays and fast key-based lookups.

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

Lysecky, R., & Vahid, F. (2018, June). *C950: Data Structures and Algorithms II*. zyBooks.

<https://learn.zybooks.com/zybook/WGUC950Template2023/chapter/6/section/1>