Author: Ian Crisp

Student ID: 010377033

Project: C950

Package Delivery Algorithm Justification

*"Algorithms + Data Structures = Programs"  
– Niklaus Wirth*

Strengths:

1. Efficiency: The algorithm used in the program aims for efficiency by trying to minimize the distance each truck has to travel. This is important because a shorter travel distance means faster delivery times and less fuel consumption.

2. Simplicity: The algorithm is relatively straightforward to understand and implement. It uses simple data structures like lists and hash tables, making it easier to debug and maintain.

Verification:

The algorithm meets all requirements set forth in the scenario, including efficient delivery of packages within a given timeframe. It also accommodates special conditions, like delivering certain packages together or within specific time slots.

Alternative Algorithms:

1. Dijkstra's Algorithm: Dijkstra's algorithm could be used to find the shortest path from the hub to all other delivery points.

2. A\* Search Algorithm: This is a more advanced path-finding algorithm that could be used to optimize the delivery route further by using heuristics.

Differences:

- Dijkstra's Algorithm: The current algorithm aims to minimize distance but doesn't necessarily find the globally optimal path. Dijkstra's algorithm would provide a more globally optimized solution by calculating the shortest paths from the source to all vertices in the given graph.

- A\* Search Algorithm: The A\* algorithm not only considers the actual cost to reach the node but also considers an additional heuristic that estimates the cost to get from that node to the goal. This could provide more optimized routes than the current algorithm.

What Would Be Done Differently:

If the project were to be done again, more advanced optimization techniques could be explored, such as genetic algorithms for route optimization. Additionally, real-time tracking and dynamic re-routing based on current conditions (e.g., traffic, weather) could be implemented for even more efficient package delivery.

Data Structure Justification:

The hash table used in the solution efficiently meets all requirements for quick insertions, deletions, and lookups, which are essential for this kind of real-time tracking and updating system.

Alternative Data Structures:

1. Balanced Search Trees: Like AVL trees or Red-Black trees could also meet the requirements. They offer logarithmic time search, insert, and delete operations.

2. Priority Queue: Could be used to keep track of the next closest package to be delivered.

Differences:

- Balanced Search Trees: These would add some overhead in maintaining the balanced property, which is not necessary for the hash table used in the solution.

- Priority Queue: This data structure would be more suitable for situations where the priority of package delivery can change dynamically. The hash table doesn't have this built-in prioritization.

Sources:

Wirth, Niklaus. "Algorithms + Data Structures = Programs." Prentice-Hall, 1976.