Duvall Roberts
C950
WGUPS Routing Program Planning

A. Algorithm Identification

Algorithm: Nearest Neighbor Algorithm
Justification: Given the constraints of minimizing distance with multiple delivery points, the nearest neighbor algorithm is ideal for efficiently creating routes by selecting the closest unvisited location (Karp, 1977).

B. Data Structure Identification

Data Structure: Hash Table
Justification: A hash table offers efficient look-up, insert, and delete operations, essential for managing real-time package data efficiently (Cormen, Leiserson, Rivest, & Stein, 2009).

B1. Relationship between Data Components

Explanation: Utilizing package IDs as keys in the hash table allows for rapid access and management of associated package details like address and delivery status, crucial for real-time updates (Cormen et al., 2009).

C. Overview of Your Program

1. Algorithm's Logic (Pseudocode)

```
function deliver_packages(packages, map, distances):
    initialize empty routes
    for each truck:
        while truck not full and packages exist:
            package = select nearest package not yet scheduled
            add package to truck's route
        calculate route distance using nearest neighbor
        if route distance acceptable:
            finalize route
        repeat until all packages scheduled
    return all routes
```

2. Programming Environment
Software: Python 3.8, Visual Studio Code
Hardware: Windows 10 PC with 16 GB RAM and an i5 Processor
Rationale: Visual Studio Code supports Python with excellent debugging tools and extensions like Python and Pylance, enhancing development productivity (Microsoft, 2021).

3. Space-Time Complexity

Complexity: O(n^2)
Explanation: The nearest neighbor algorithm has a quadratic time complexity, as each node (package) is compared to every other node to find the nearest one. This results in O(n^2) time complexity, where n is the number of packages. This complexity is manageable for the moderate size of the dataset in this project but may become inefficient for very large datasets (Karp, 1977).
Hash Table Complexity:
Time Complexity: O(1) for average-case insertions, deletions, and look-ups due to direct access via hashing. However, the worst-case time complexity can be O(n) if many collisions occur, necessitating handling (Cormen et al., 2009).
Space Complexity: O(n) for storing n packages, where each package ID and its associated details are kept in the hash table.

4. Scalability and Adaptability
Explanation: The program can scale by modifying parameters like the number of trucks or adjusting the hash table to manage more packages efficiently (Cormen et al., 2009).

5. Efficiency and Maintenance
Advantages: Simple debugging and clear algorithm structure due to the intuitive use of hash tables.
Disadvantages: Scalability concerns with the O(n^2) complexity of the nearest neighbor approach, especially as the number of delivery points grows significantly.

6. Strengths and Weaknesses of Hash Table
Strengths: Rapid access times and ease of implementation.
Weaknesses: Space complexity can degrade performance during extensive rehashing phases if not managed correctly (Cormen et al., 2009).

7. Key Choice Justification
Chosen Key: Package ID
Justification: Ensures unique identification and efficient retrieval, which are crucial for real-time updates on delivery statuses (Cormen et al., 2009).

References:

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). MIT Press.
- Karp, R. M. (1977). Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane. Mathematics of Operations Research, 2(3), 209-224.

- Microsoft. (2021). Visual Studio Code. [Visual Studio Code] https://code.visualstudio.com/