C950 WGUPS Algorithm Overview

Linda Vataksi

ID # 011273373

WGU Email: lvataks@wgu.edu

Date: 02/28/ 2024

C950 Data Structures and Algorithms II

TASK 2: WGUPS ROUTING PROGRAM PLANNING

**Part D:**

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.

8:45 AM:

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

1. 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.

9:45 AM:

A screenshot of a computer screen

Description automatically generated

A screen shot of a computer

Description automatically generated

1. 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.

12:45 PM:

A screenshot of a computer screen

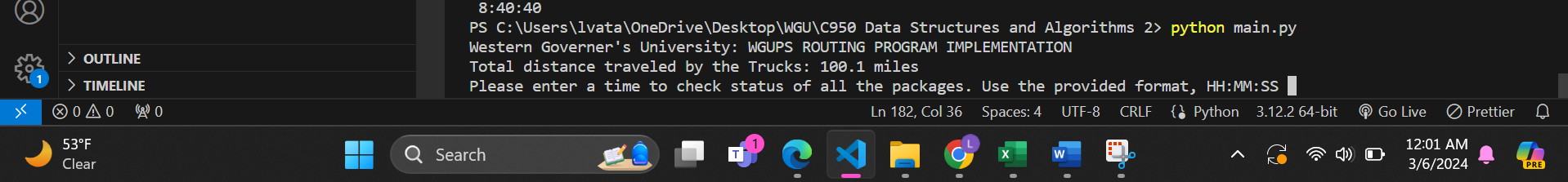
Description automatically generated

A screen shot of a computer screen

Description automatically generated

**Part E**

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



**Part 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.

Two advantages of the nearest neighbor algorithm include its simplicity and efficiency in solving optimization problems, such as the traveling salesman problem. Its straightforward approach of iteratively selecting the nearest available neighbor from the current location makes it easy to understand and implement, requiring minimal computational resources. Despite its simplicity, the algorithm yields good solutions quickly, especially for large datasets. By selecting locally optimal solutions at each step, it can efficiently navigate through complex problem spaces, making it particularly suitable for real-time applications or scenarios with resource constraints where quick decision-making is essential. Overall, the nearest neighbor algorithm's combination of simplicity and efficiency makes it a valuable tool in various optimization contexts.

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

The nearest neighbor algorithm fulfills all specified criteria in this scenario by ensuring timely delivery of all packages, maintaining a total mileage below 140 miles, and accommodating special delivery instructions. It adjusts departure times for packages arriving at the hub at 9:05 accordingly and arranges deliveries with other packages when necessary. Importantly, the algorithm continues until all packages are successfully delivered.

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

Two alternative algorithms that could meet the requirements of the scenario are Dijkstra's algorithm and the Greedy algorithm.

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

Dijkstra's algorithm is a graph-based algorithm used to find the shortest path from a source node to all other nodes in a weighted graph. Unlike the nearest neighbor algorithm, which is specifically designed for solving the traveling salesman problem, Dijkstra's algorithm can handle more general routing and pathfinding problems. It operates by iteratively selecting the node with the shortest distance from the source and updating the distances to its neighboring nodes accordingly [2]. Dijkstra's algorithm guarantees finding the shortest paths in non-negative weighted graphs, making it suitable for scenarios where minimizing distance or time is a priority.

Unlike Dijkstra's algorithm, which considers the shortest path from a source node to all other nodes, the Greedy algorithm focuses on making immediate decisions based on local information without considering the overall structure of the problem. While Greedy algorithms can be straightforward to implement and computationally efficient, they are not always the most efficient. “The choice of option does not consider additional subsequent options and may or may not lead to an optimal solution [1]”. Despite this drawback, I believe the greedy algorithm would have still been able to meet the requirements of this scenario.

**Part 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.

If I could do this project differently, one thing I would do is try to further optimize the truck loading. In the current version of my project, I manually loaded the trucks. In a future edition, I would like to optimize the truck loading with an algorithm. One potential algorithm I could try to implement for this optimization could be the greedy algorithm.

**Part H**

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

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

Two other data structures that could meet the same requirements include arrays and binary search trees.

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

Arrays and binary search trees (BSTs) are two alternative data structures that could meet the requirements of the scenario. Arrays provide direct access to elements by index, making them simple to implement and memory-efficient for small to medium-sized datasets. However, they have a fixed size and may waste memory for unused indices. On the other hand, BSTs maintain an ordered structure, allowing efficient searching, insertion, and deletion operations based on package IDs. While BSTs offer ordering capabilities and are suitable for scenarios where packages need to be accessed in a specific order, they may have higher memory overhead and slower average-case lookup times compared to hash tables.

# I. Sources - Works Cited

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

Retrieved March 22, 2021, from <https://learn.zybooks.com/zybook/WGUC950AY20182019/>

2. *Understanding dijkstra algorithm: History, working, advantages, disadvantages, Applications & Complexity*. Testbook. (n.d.). https://testbook.com/gate/dijkstra-algorithm-notes