C950 Task-2 WGUPS Write-Up

(Task-2: The implementation phase of the WGUPS Routing Program).

(Zip your source code and upload it with this file)

Sarah Kouhail

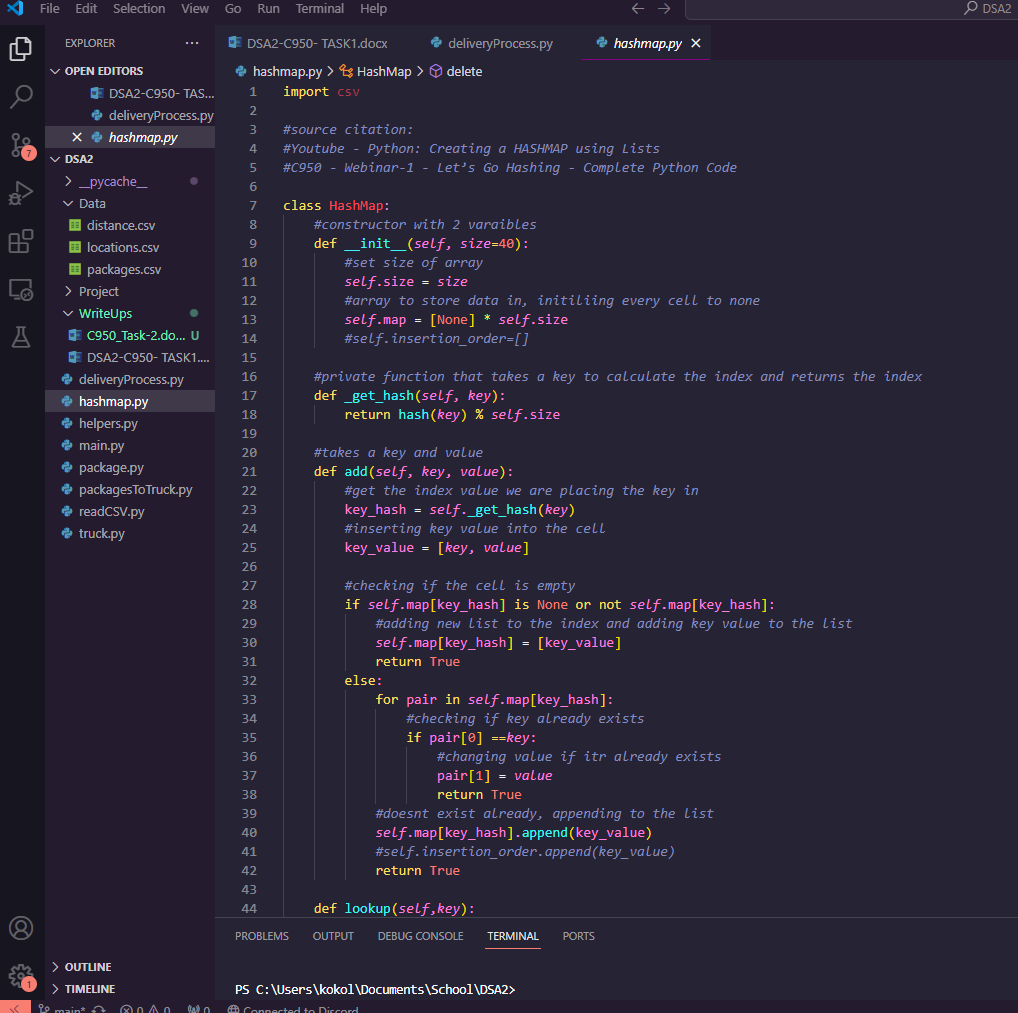
ID #010604128

WGU Email: skouhai@wgu.edu

03/07/2024

C950 Data Structures and Algorithms II

# A. Hash Table



A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer screen shot of text

Description automatically generated

# B. Look-Up Functions

# A computer screen shot of a code Description automatically generated

# C. Original Code

A screenshot of a computer program

Description automatically generated

A computer screen shot of text

Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer screen with text

Description automatically generated

A screenshot of a computer program

Description automatically generated

# C1. Identification Information

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

# C2. Process and Flow Comments

# A screenshot of a computer program Description automatically generated

# A computer screen shot of text Description automatically generated

A computer screen shot of text

Description automatically generated

# D. Interface

Interface screenshot goes here

A screen shot of a computer

Description automatically generated

# D1. First Status Check

Screen shot goes here

A screenshot of a computer

Description automatically generated

A screen shot of a computer screen

Description automatically generated

# D2. Second Status Check

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

# D3. Third Status Check

Screenshot goes here

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

# E. Screenshot of Code Execution

Screenshot goes here

A screenshot of a computer

Description automatically generated

# F1. Strengths of the Chosen Algorithm

For this project, I chose the nearest neighbor algorithm. One strength of this algorithm is that is it efficient. It sorts the packages based on their distance from one another, allowing the delivery process to minimize the total distance traveled by each truck. Another strength of this algorithm is that it is simplistic. The implementation is straightforward. It iterates through each package and selects the next nearest package at each stop. Overall, it makes the delivery process easier to digest.

# F2. Verification of Algorithm

This algorithm meets all the requirements because it constructs a solution by repeatedly selecting the nearest unvisited vertex(location) from the current vertex. This optimizes the delivery route by iterating over all packages and adding them into a list sorted by delivery order, based on the distance between each package. It also updates the truck progress, handles error cases, and ensures timely delivery based on the package constraints.

# F3. Other Possible Algorithms

Two other algorithms that could be used for this solution would be Dijkstra’s algorithm and the A-star algorithm. Dijkstra’s algorithm is used to find the shortest path from a single source vertex to all other vertices in a weighted map. As for A-star, it is a heuristic search algorithm that uses functions to estimate the distance of reaching from each vertex. This would help guide the search for the most efficient route of optimization regarding the solution of this project.

# F3a. Algorithm Differences

Unlike nearest neighbor, which focuses on finding the nearest unvisited vertex, Dijkstra’s algorithm takes into consideration all possible vertices to find the shortest path to each one. As for A-star, it is a heuristic search algorithm that uses functions to estimate the distance of reaching from each vertex. Unlike the algorithm implemented in the solution, the A-star algorithm takes into consideration the overall picture to find the optimal path. The algorithm used for the solution (nearest neighbor) only selects the nearest package at each step.

# G. Different Approach

If I did this project again with less of a time constraint, I would want to find a more efficient way to determine which packages should go on which truck depending on not only their destinations but constraints. I created a file that outputs lists of packages with the same addresses as well as grouping the packages up by deadline time. I used this manually to determine package placements, however, I would have liked to make some sort of function that also output the next nearest location to the packages with the same addresses. When manually entering the packages, I just moved some around to determine which would be the best fit. Another modification I would make would be to have the user enter a time after they make an option from the interface rather than before.

# H. Verification of Data Structure

The data structure used in this solution is a HashMap. This was used to store the package information keyed by the package ID. Before implementing the algorithm, a HashMap can be constructed considering the package information provided in the Excel sheets found in the task. Each package can be given a unique ID, which will act as its key, while the object itself serves as the value. Using a HashMap also allows for efficient retrieval. It provides a constant-time average-case complexity, making it efficient to quickly access package information based on a unique ID. With collision handling, it ensures that different packages with the potential to have identical hash values can exist in the same HashMap. Finally, it also allows for dynamic updating of the data loaded into it. As packages are updated (deleted, added, etc.), it can adjust its internal structure to remain efficient for data retrieval.

# H1. Other Data Structures

A different data structure I could have used for the solution would be an array. It could be used to store the package objects while the index of the array could represent the package ID. This would also offer constant-time access. Another data structure that could meet the requirements of this project would be linked lists. Each node could represent an object and the package ID would be the unique identifier here as well. Linked lists provide efficient insertion and deletion operations as well.

# H1a. Data Structure Differences

The difference between the data structure used for this solution and an array is that an array would require more memory. Memory usage of an array in this situation could lead to space inefficiency, especially if package IDs are sparse. The difference between the HashMap and the linked list would be that the linked list may not offer constant-time access. Traversing through a linked list to find a specific package could lead to linear complexity in the worst-case.

# I. Sources

Text goes here

An example:

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

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

# J. Professional Communication