

C950 Data Structures & Algorithms II

Introduction

For Tasks 1 and 2, you will apply the algorithms and data structures you studied in this course to solve a real programming problem. You will also implement an algorithm to route delivery trucks that will allow you to meet all delivery constraints while traveling under 140 miles. You will then describe and justify the decisions you made while creating this program.

The skills you showcase in your completed project may be useful in responding to technical interview questions for future employment. This project may also be added to your portfolio to show to future employers.

Scenario

The Western Governors University Parcel Service (WGUPS) needs to determine an efficient route and delivery distribution for their daily local deliveries (DLD) because packages are not currently being consistently delivered by their promised deadline. The Salt Lake City DLD route has three trucks, two drivers, and an average of 40 packages to deliver each day. Each package has specific criteria and delivery requirements that are listed in the attached “WGUPS Package File.”

Your task is to determine an algorithm, write code, and present a solution where all 40 packages will be delivered on time while meeting each package’s requirements and keeping the combined total distance traveled under 140 miles for all trucks. The specific delivery locations are shown on the attached “Salt Lake City Downtown Map,” and distances to each location are given in the attached “WGUPS Distance Table.” The intent is to use the program for this specific location and also for many other cities in each state where WGU has a presence. As such, you will need to include detailed comments to make your code easy to follow and to justify the decisions you made while writing your scripts.

The supervisor should be able to see, at assigned points, the progress of each truck and its packages by any of the variables listed in the “WGUPS Package File,” including what has been delivered and at what time the delivery occurred.

Assumptions

- Each truck can carry a maximum of 16 packages, and the ID number of each package is unique.
- The trucks travel at an average speed of 18 miles per hour and have an infinite amount of gas with no need to stop.

- There are no collisions.
 - Three trucks and two drivers are available for deliveries. Each driver stays with the same truck as long as that truck is in service.
 - Drivers leave the hub no earlier than 8:00 a.m., with the truck loaded, and can return to the hub for packages if needed.
 - The delivery and loading times are instantaneous (i.e., no time passes while at a delivery or when moving packages to a truck at the hub). This time is factored into the calculation of the average speed of the trucks.
 - There is up to one special note associated with a package.
 - The delivery address for package #9, Third District Juvenile Court, is wrong and will be corrected at 10:20 a.m. WGUPS is aware that the address is incorrect and will be updated at 10:20 a.m. However, WGUPS does not know the correct address (410 S. State St., Salt Lake City, UT 84111) until 10:20 a.m.
 - The distances provided in the “WGUPS Distance Table” are equal regardless of the direction traveled.
 - The day ends when all 40 packages have been delivered.
-

Task 1

A.

Identify a named self-adjusting algorithm (e.g., nearest neighbor algorithm, greedy algorithm) that could be used to create your program to deliver the packages.

B.

Identify a self-adjusting data structure, such as a hash table, that could be used with the algorithm identified in part A to store the package data.

1. Explain how your data structure accounts for the relationship between the data components you are storing.

C.

Write an overview of your program in which you do the following:

1. Explain the algorithm’s logic using pseudocode.

Note: You may refer to the attached “Sample Core Algorithm Overview” to complete part

C1.

2. Describe the programming environment you will use to create the Python application, including both the software and hardware you will use.
3. Evaluate the space-time complexity of each major segment of the program and the entire program using big-O notation.
4. Explain the capability of your solution to scale and adapt to a growing number of packages.
5. Discuss why the software design would be efficient and easy to maintain.
6. Describe both the strengths and weaknesses of the self-adjusting data structure (e.g., the hash table).
7. Justify the choice of a key for efficient delivery management from the following components:
 - delivery address
 - delivery deadline
 - delivery city
 - delivery zip code
 - package ID
 - package weight
 - delivery status (i.e., at the hub, en route, or delivered), including the delivery time

D.

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

E.

Demonstrate professional communication in the content and presentation of your submission.

Task 2

A.

Develop a hash table, without using any additional libraries or classes, that has an insertion function that takes the package ID as input and inserts each of the following data components into the hash table:

- delivery address
- delivery deadline
- delivery city
- delivery zip code
- package weight
- delivery status (i.e., at the hub, en route, or delivered), including the delivery time

B.

Develop a look-up function that takes the package ID as input and returns each of the following corresponding data components:

- delivery address
- delivery deadline
- delivery city
- delivery zip code
- package weight
- delivery status (i.e., at the hub, en route, or delivered), including the delivery time

C.

Write an original program that will deliver all packages and meet all requirements using the attached supporting documents “Salt Lake City Downtown Map,” “WGUPS Distance Table,” and “WGUPS Package File.”

1. Create an identifying comment within the first line of a file named “main.py” that includes your student ID.
2. Include comments in your code to explain both the process and the flow of the program.

D.

Provide an intuitive interface for the user to view the delivery status (including the delivery time) of any package at any time and the total mileage traveled by all trucks. (The delivery status should report the package as at the hub, en route, or delivered. Delivery status must include the time.)

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.
2. 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.
3. 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.

E.

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

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.
2. Verify that the algorithm used in the solution meets all requirements in the scenario.
3. Identify two other named algorithms that are different from the algorithm implemented in the solution and would meet all requirements in the scenario.
 - Describe how both algorithms identified in part F3 are different from the algorithm used in the solution.

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.

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.

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

I.

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

J.

Demonstrate professional communication in the content and presentation of your submission.