NHP3 — NHP3 TASK 1: WGUPS ROUTING PROGRAM PLANNING

DATA STRUCTURES AND ALGORITHMS II — C950

PRFA — NHP3

COMPETENCIES

4048.5.3 : Dictionaries and Sets

The graduate incorporates dictionaries and sets in order to organize data into key-value pairs.

4048.5.4 : Self-Adjusting Data Structures

The graduate evaluates the space and time complexity of self-adjusting data structures using big-O notation to improve the performance of applications.

4048.5.6 : NP-Completeness and Turing Machines

The graduate evaluates computational complexity theories in order to apply models to specific scenarios.

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

This task is the planning phase of the WGUPS Routing Program.

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 two of the 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.

REQUIREMENTS

Your submission must be your original work. No more than a combined total of 30% of the submission and no more than a 10% match to any one individual source can be directly quoted or closely paraphrased from sources, even if cited correctly. The similarity report that is provided when you submit your task can be used as a guide.

You must use the rubric to direct the creation of your submission because it provides detailed criteria that will be used to evaluate your work. Each requirement below may be evaluated by more than one rubric aspect. The rubric aspect titles may contain hyperlinks to relevant portions of the course.

*Tasks may **not** be submitted as cloud links, such as links to Google Docs, Google Slides, OneDrive, etc., unless specified in the task requirements. All other submissions must be file types that are uploaded and submitted as attachments (e.g., .docx, .pdf, .ppt).*

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.

File Restrictions

File name may contain only letters, numbers, spaces, and these symbols: ! - _ . * ' ( )
File size limit: 400 MB
File types allowed: doc, docx, rtf, xls, xlsx, ppt, pptx, odt, pdf, txt, qt, mov, mpg, avi, mp3, wav, mp4, wma, flv, asf, mpeg, wmv, m4v, svg, tif, tiff, jpeg, jpg, gif, png, zip, rar, tar, 7z

RUBRIC

A. :ALGORITHM IDENTIFICATION

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A named self-adjusting algorithm is not identified.		

| | The named self-adjusting algorithm identified would not be appropriate to use to create a program to deliver the packages. | The named self-adjusting algorithm identified would be appropriate to use to create a program to deliver the packages. |

B. :DATA STRUCTURE IDENTIFICATION

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A self-adjusting data structure is not identified.	The self-adjusting data structure identified would not be appropriate to use with the algorithm identified in part A to store the package data.	The self-adjusting data structure identified would be appropriate to use with the algorithm identified in part A to store the package data.

B1. :EXPLANATION OF DATA STRUCTURE

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
An explanation about how the data structure accounts for the relationship between the data components being stored is not provided.	The explanation is missing details or does not accurately describe how the data structure accounts for the relationship between the data components being stored.	The explanation thoroughly and accurately describes how the data structure accounts for the relationship between the data components being stored.

C1. :ALGORITHM'S LOGIC

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
An explanation in pseudocode is not provided.	The explanation in pseudocode does not accurately describe the algorithm's logic.	The explanation in pseudocode accurately describes the algorithm's logic.

C2. :DEVELOPMENT ENVIRONMENT

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT

| A description of the programming environment is not provided. | The description does not accurately explain the programming environment that will be used to create the Python application. Or the description is missing an explanation of the software or hardware that will be used. | The description accurately explains the programming environment that will be used to create the Python application. The description includes *both* the software and hardware that will be used. |

C3. :SPACE-TIME COMPLEXITY USING BIG-O NOTATION

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
An evaluation of the space-time complexity using big-O notation is not provided.	The evaluation does not use big-O notation, or it does not accurately determine the space-time complexity of 1 or more major segments of the program. Or an evaluation is missing for space, time, or the entire program.	The evaluation uses big-O notation and accurately determines the space-time complexity of *each* major segment of the program and the entire program.

C4. :SCALABILITY AND ADAPTABILITY

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
An explanation of the capability of the solution to scale and adapt to a growing number of packages is not provided.	The explanation is missing details, or it does not accurately describe the capability of the solution to scale and adapt to a growing number of packages.	The explanation thoroughly and accurately describes the capability of the solution to scale and adapt to a growing number of packages.

C5. :SOFTWARE EFFICIENCY AND MAINTAINABILITY

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A discussion about why the software design would be efficient and easy to maintain is not provided.	The discussion does not accurately explain why the software design would be efficient or why it would be easy to maintain.	The discussion accurately explains why the software design would be efficient and easy to maintain.

C6. :SELF-ADJUSTING DATA STRUCTURES

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT

A description of the strengths and weaknesses of the self-adjusting data structure is not provided.	The description is missing details, or it does not accurately explain *both* the strengths and weaknesses of the self-adjusting data structure. Or a description is missing for the strengths or the weaknesses.	The description thoroughly and accurately explains *both* the strengths and weaknesses of the self-adjusting data structure.

C7. :DATA KEY

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A justification of the key choice is not provided.	The justification of the key choice is provided, but it does not appropriately address efficient delivery management.	The justification of the key choice appropriately addresses efficient delivery management.

D. :SOURCES

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not include both in-text citations and a reference list for sources that are quoted, paraphrased, or summarized.	The submission includes in-text citations for sources that are quoted, paraphrased, or summarized and a reference list; however, the citations or reference list is incomplete or inaccurate.	The submission includes in-text citations for sources that are properly quoted, paraphrased, or summarized and a reference list that accurately identifies the author, date, title, and source location as available.

E. :PROFESSIONAL COMMUNICATION

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
Content is unstructured, is disjointed, or contains pervasive errors in mechanics, usage, or grammar. Vocabulary or tone is unprofessional or distracts from the topic.	Content is poorly organized, is difficult to follow, or contains errors in mechanics, usage, or grammar that cause confusion. Terminology is misused or ineffective.	Content reflects attention to detail, is organized, and focuses on the main ideas as prescribed in the task or chosen by the candidate. Terminology is pertinent, is used correctly, and effectively conveys the intended meaning. Mechanics, usage, and grammar promote accurate interpretation and understanding.

SUPPORTING DOCUMENTS

Sample Core Algorithm Overview.docx

SLC downtown map.docx

WGUPS Distance Table.xlsx

WGUPS Package File.xlsx