NHP1 — NHP1 TASK 1: WGUPS ROUTING PROGRAM

DATA STRUCTURES AND ALGORITHMS II — C950
PRFA — NHP1

COMPETENCIES

4048.5.1: : Non-Linear Data Structures
The graduate creates software applications that incorporate non-linear data structures for efficient and maintainable software.

4048.5.2: : Hashing Algorithms and Structures
The graduate writes code using hashing techniques within an application to perform searching operations.

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.

4048.5.5: : Self-Adjusting Heuristics
The graduate writes code using self-adjusting heuristics 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 this assessment, you will apply the algorithms and data structures you studied in this course to solve a real programming problem. You will implement an algorithm to route delivery trucks that will allow you to meet all delivery deadlines while traveling the least number of miles. You will also 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 the best 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.

Your task is to determine the best algorithm, write code, and present a solution where all 40 packages, listed in the attached “WGUPS Package File,” will be delivered on time with the least number of miles added to the combined mileage total of 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.”

While you work on this assessment, take into consideration the specific delivery time expected for each package and the possibility that the delivery requirements—including the expected delivery time—can be changed by management at any time and at any point along the chosen route. In addition, you should keep in mind that 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 what time the delivery occurred.

The intent is to use this solution (program) for this specific location and to use the same program in many cities in each state where WGU has a presence. As such, you will need to include detailed comments, following the industry-standard Python style guide, to make your code easy to read and to justify the decisions you made while writing your program.

Assumptions:

- Each truck can carry a maximum of 16 packages.
- Trucks travel at an average speed of 18 miles per hour.
- Trucks have a “infinite amount of gas” with no need to stop.
- Each driver stays with the same truck as long as that truck is in service.
- Drivers leave the hub at 8:00 a.m., with the truck loaded, and can return to the hub for packages if needed. The day ends when all 40 packages have been delivered.
- Delivery time is instantaneous, i.e., no time passes while at a delivery (that time is factored into the average speed of the trucks).
- There is up to one special note for each package.
- The wrong delivery address for package #9, *Third District Juvenile Court*, will be corrected at 10:20 a.m. The correct address is 410 S State St., Salt Lake City, UT 84111.
- The package ID is unique; there are no collisions.
- No further assumptions exist or are allowed.

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. An originality report is provided when you submit your task that 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.

Section 1: Programming/Coding

A. Identify the algorithm that will be used to create a program to deliver the packages and meets *all* requirements specified in the scenario.

B. Write a core algorithm overview, using the sample given, in which you do the following:
  1. Comment using pseudocode to show the logic of the algorithm applied to this software solution.
  2. Apply programming models to the scenario.
  3. Evaluate space-time complexity using Big O notation throughout the coding and for the entire program.
  4. Discuss the ability of your solution to adapt to a changing market and to scalability.
  5. Discuss the efficiency and maintainability of the software.
  6. Discuss the self-adjusting data structures chosen and their strengths and weaknesses based on the scenario.

C. Write an original code to solve and to meet the requirements of lowest mileage usage and having *all* packages delivered on time.
  1. Create a comment within the first line of your code that includes your first name, last name, and student ID.
  2. Include comments at *each* block of code to explain the process and flow of the coding.

D. Identify a data structure that can be used with your chosen algorithm to store the package data.
  1. Explain how your data structure includes the relationship between the data points you are storing.

Note: Do NOT use any existing data structures. You must design, write, implement, and debug all *code that you turn in for this assessment. Code downloaded from the internet or acquired from another student or any other source may not be submitted and will result in automatic failure of this assessment.*

E. Develop a hash table, without using any additional libraries or classes, with an insertion function that takes the following components as input and inserts the components into the hash table:
  - package ID number
  - delivery address
  - delivery deadline
  - delivery city
  - delivery zip code
  - package weight
  - delivery status (e.g., delivered, in route)

F. Develop a look-up function that takes the following components as input and returns the corresponding data elements:
- package ID number
- delivery address
- delivery deadline
- delivery city
- delivery zip code
- package weight
- delivery status (e.g., delivered, in route)

G. Provide an interface for the insert and look-up functions to view the status of any package at any time. This function should return *all* information about *each* package, including delivery status.
1. Provide screenshots to show package status of *all* packages at a time between 8:35 a.m. and 9:25 a.m.
2. Provide screenshots to show package status of *all* packages at a time between 9:35 a.m. and 10:25 a.m.
3. Provide screenshots to show package status of *all* packages at a time between 12:03 p.m. and 1:12 p.m.

H. Run your code and provide screenshots to capture the complete execution of your code.

Section 2: Annotations

I. Justify your choice of algorithm by doing the following:
1. Describe *at least* **two** strengths of the algorithm you chose.
2. Verify that the algorithm you chose meets *all* the criteria and requirements given in the scenario.
3. Identify **two** other algorithms that could be used and would have met the criteria and requirements given in the scenario.
    a. Describe how *each* algorithm identified in part I3 is different from the algorithm you chose to use in the solution.

J. Describe what you would do differently if you did this project again.

K. Justify your choice of data structure by doing the following:
1. Verify that the data structure you chose meets *all* the criteria and requirements given in the scenario.
    a. Describe the efficiency of the data structure chosen.
    b. Explain the expected overhead when linking to the next data item.
    c. Describe the implications of when more package data is added to the system or other changes in scale occur.
2. Identify **two** other data structures that can meet the same criteria and requirements given in the scenario.
    a. Describe how *each* data structure identified in part K2 is different from the data structure you chose to use in the solution.

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

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

RUBRIC

A:ALGORITHM SELECTION

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
An algorithm is not provided.	The identified algorithm is not capable of performing the task or is not capable of performing the task while meeting all the requirements outlined in the scenario.	The identified algorithm can perform the task and meets all requirements as outlined in the scenario.

B1:LOGIC COMMENTS

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
Comments are not provided.	The comments either do not align to industry standards or the logic applied to the solution contains inaccuracies.	The comments align to industry standards and accurately explain the logic applied to the solution.

B2:APPLICATION OF PROGRAMMING MODELS

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
An application of programming models is not provided.	The application of programming models either does not include the communication protocol that is used to exchange data; does not include the target host environment used to host the server applica-	The application of programming models includes the communication protocol that is used to exchange data; the target host environment used to host the server application program; and the interaction semantics defined by the application to con-

	tion program; or does not include the interaction semantics defined by the application to control connect, data exchange, and disconnect sequences. Or the models contain inaccuracies.	trol connect, data exchange, and disconnect sequences.

B3:SPACE-TIME AND BIG-O

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
An evaluation of the space-time complexity is not provided.	The evaluation uses Big O notation to show space-time complexity but does not correctly quantify the amount of time taken by their chosen algorithm to run, or the evaluation does not consider the length of the string representing the input in calculating the complexity.	The evaluation shows the space-time complexity using Big O notation for each block of coding and for the entire program.

B4:ADAPTABILITY

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A discussion of scalability is not provided.	The discussion does not include the chosen algorithm's ability to handle a growing amount of work or its scalability to accommodate growth. Or the discussion contains inaccuracies.	The discussion includes the chosen algorithm's ability to handle a growing amount of work and its scalability to accommodate growth.

B5:SOFTWARE EFFICIENCY AND MAINTAINABILITY

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A discussion of software effi-	A discussion of the software's	A discussion of the software's

ciency is not provided.

efficiency is provided, but the discussion does not address why the software is efficient or does not address why the software is easy to maintain. Or the discussion contains inaccuracies.

efficiency is provided, and the discussion addresses how the software is efficient and easy to maintain.

B6:SELF-ADJUSTING DATA STRUCTURES

NOT EVIDENT

A discussion of self-adjusting data structures is not provided.

APPROACHING COMPETENCE

The discussion of self-adjusting data structures includes the ability of the data structure to adapt when accessed but not how that adaption affects running time. Or the discussion contains inaccuracies.

COMPETENT

The discussion of self-adjusting data structures includes the ability of the data structure to adapt when accessed and how that adaptation affects running time.

C:ORIGINAL CODE

NOT EVIDENT

The code provided is not original, or no code is provided.

APPROACHING COMPETENCE

The original code runs properly but does not deliver all packages on time or does not add the least number to the combined mileage total of all trucks.

COMPETENT

The original code runs properly and delivers all packages on time while adding the least number to the combined mileage total of all trucks.

C1:IDENTIFICATION INFORMATION

NOT EVIDENT

An initial comment with information about the candidate's identity is not provided.

APPROACHING COMPETENCE

The initial comment contains some identification information for the candidate, but it is not located within the first line of code, or it does not include

COMPETENT

The initial comment is located within the first line of code and includes the candidate's first name, last name, and student ID.

the candidate's first name, the candidate's last name, or the candidate's student ID.

C2:PROCESS AND FLOW COMMENTS

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
Comments are not provided.	Comments are found within the coding, but the comments are not at each large block of code, or they do not improve the readability of the code or show the intent and decisions made while developing the program. Or the comments contain inaccuracies.	Comments are found within the coding at each large block of code, improve the readability of the code, and show the intent and decisions made while developing the program.

D:DATA STRUCTURE

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not identify a data structure.	The submission identifies a data structure, but the data structure would not perform well when applied to the usage described in the scenario.	The submission identifies a data structure that performs well when applied to the usage described in the scenario.

D1:EXPLANATION OF DATA STRUCTURE

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not provide an explanation of the data structure.	The submission explains the data structure but does not explain the relationship between the data points to be stored. Or the explanation contains inaccuracies.	The submission accurately explains the data structure and how that data structure accounts for the relationship between the data points to be stored.

E:HASH TABLE

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A hash table is not provided.	The hash table does not have an insertion function or has an insertion function that does not account for all of the given components. Or the table contains errors.	The hash table has an insertion function that includes, as input, all of the given components.

F:LOOK-UP FUNCTION

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A look-up function is not provided.	The look-up function does not include all of the given data elements, the look-up function does not complete searches and return correct data, or the look-up function does not list the status of all packages. Or the table contains errors.	The look-up function includes all of the given data elements, completes searches and returns correct data, and lists the status of all packages.

G:INTERFACE

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
An interface is not provided.	The interface includes simple access to the program, but the information included is insufficient for a user to communicate or use the program. Or the interface contains errors.	The interface includes the information needed for a user to communicate and use the program.

G1:FIRST STATUS CHECK

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A screenshot is not provided.	The screenshots show an in-	The screenshots show a listing of all packages that are loaded

	complete listing of the packages that are loaded on each truck, or they do not show the current status of each package at a time between 8:35 a.m. and 9:25 a.m.	on each truck and the current status of each package at a time between 8:35 a.m. and 9:25 a.m.

G2:SECOND STATUS CHECK

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A screenshot is not provided.	The screenshots show an incomplete listing of the packages that are loaded on each truck, or they do not show the current status of each package at a time between 9:35 a.m. and 10:25 a.m.	The screenshots show a listing of all packages that are loaded on each truck and the current status of each package at a time between 9:35 a.m. and 10:25 a.m.

G3:THIRD STATUS CHECK

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A screenshot is not provided.	The screenshots show an incomplete listing of the packages that are loaded on each truck, or they do not show the current status of each package at a time between 12:03 p.m. and 1:12 p.m.	The screenshots show a listing of all packages that are loaded on each truck and the current status of each package at a time between 12:03 p.m. and 1:12 p.m.

H:SCREENSHOTS OF CODE EXECUTION

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
Screenshots are not provided.	The screenshots capture an incomplete execution of the code.	The screenshots capture a complete execution of the code.

I1:STRENGTHS OF THE CHOSEN ALGORITHM

NOT EVIDENT

A description is not provided.

APPROACHING COMPETENCE

The description includes at least 2 strengths of the chosen algorithm, but they are not related to the scenario. Or the strengths relate to the scenario, but they do not match the chosen algorithm. Or the description contains inaccuracies.

COMPETENT

The description includes at least 2 specific strengths of the chosen algorithm as they apply to the scenario.

I2:VERIFICATION OF ALGORITHM

NOT EVIDENT

A verification is not provided.

APPROACHING COMPETENCE

The verification includes the total miles added to all trucks, but the total is not the least number of miles possible or does not state that all packages were delivered on time.

COMPETENT

The verification includes the total miles added to all trucks, and it states that all packages were delivered on time.

I3:OTHER POSSIBLE ALGORITHMS

NOT EVIDENT

The submission does not identify any other algorithms.

APPROACHING COMPETENCE

The submission identifies 2 other algorithms, but 1 or more do not meet the criteria given in the scenario.

COMPETENT

The submission identifies 2 other algorithms that could meet the requirements of the scenario.

I3A:ALGORITHM DIFFERENCES

NOT EVIDENT

A description is not provided.

APPROACHING COMPETENCE

The description does not in-

COMPETENT

The description includes attrib-

	clude attributes of each algorithm identified in part I3, or the description fails to compare the identified attributes to the attributes of the algorithm used in the solution. Or the description contains inaccuracies.	utes of each algorithm identified in part I3 and how the identified attributes compare to the attributes of the algorithm used in the solution.

J:DIFFERENT APPROACH

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A description is not provided.	The description identifies at least 1 aspect of the process the candidate could change, but does not describe how the candidate would modify the process.	The description includes at least 1 aspect of the process that the candidate would do differently and includes how the candidate would modify the process.

K1:VERIFICATION OF DATA STRUCTURE

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A verification is not provided.	The verification shows only a portion of the criteria has been met: either the least number of total miles were added to all trucks, all packages were not delivered on time, the hash table with look-up function is missing, or the reporting needed is inaccurate or inefficient.	The verification shows all the criteria has been met: the least number of total miles added to all trucks, all packages were delivered on time, the hash table with look-up function is present, and the reporting needed is accurate and efficient.

K1A:EFFICIENCY

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A description of the efficiency of the chosen data structure is	The description of the efficiency of the data structure used in	The description of the efficiency of the data structure used in the

not provided.	the solution does not include both what type of data is being used and how that data is being used.	solution includes what type of data is being used and how that data is being used.

K1B:OVERHEAD

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
An explanation is not provided.	The explanation does not include all three topics of: computational time, memory, and bandwidth when handling data in this program.	The explanation includes the computational time, memory, and bandwidth aspects when handling data in this program.

K1C:IMPLICATIONS

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A description of the changes needed due to scale is not provided.	The description does not address the changes needed when the number of packages, the number of trucks, or the number of cities increases. The description does not address the idea of control when different or numerous sub-applications or subsystems are incorporated through the expansion to numerous cities.	The description addresses the changes needed when the number of packages, the number of trucks, and the number of cities increase. The description addresses the idea of control when different or numerous sub-applications or subsystems are incorporated through the expansion to numerous cities.

K2:OTHER DATA STRUCTURES

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not identify other data structures.	The submission identifies only 1 data structure other than the one used in the solution, or the submission identifies 2 data structures, but 1 or more of the	The submission identifies 2 data structures other than the one used in the solution that meet the criteria and requirements in the scenario.

data structures do not meet the criteria and requirements in the scenario.

K2A:DATA STRUCTURE DIFFERENCES

NOT EVIDENT
A description of the differences is not provided.

APPROACHING COMPETENCE
The description does not include attributes of each data structure identified in part K2, or the description fails to compare these attributes to the attributes of the data structure used in the solution.

COMPETENT
The description includes the attributes of each data structure identified in part K2 and compares these attributes to the attributes of the data structure used in the solution.

L:SOURCES

NOT EVIDENT
The submission does not include both in-text citations and a reference list for sources that are quoted, paraphrased, or summarized.

APPROACHING COMPETENCE
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.

COMPETENT
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.

M:PROFESSIONAL COMMUNICATION

NOT EVIDENT
Content is unstructured, is disjointed, or contains pervasive errors in mechanics, usage, or grammar. Vocabulary or tone is unprofessional or distracts from the topic.

APPROACHING COMPETENCE
Content is poorly organized, is difficult to follow, or contains errors in mechanics, usage, or grammar that cause confusion. Terminology is misused or ineffective.

COMPETENT
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