Abdellah Zakani

Student ID:
009970024

A. Identify a named self-adjusting algorithm (e.g., "Nearest Neighbor algorithm," "Greedy
    algorithm") that you used to create your program to deliver the packages.

A. The self-adjusting algorithm is the nearest neighbor algorithm.

B. 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 B1.
B.
    1. For each package not delivered
            If the distance between the truck and the package address <= the next package address
                Next address=distance
                Next package=package
      Add the package to truck
      Remove package from not delivered [ ]
      Add mileage to truck
      Update truck address
      Updating next package delivery and update times

    2. Describe the programming environment you used to create the Python application.
    2. I used the PyCharm community edition, python interpreter version 3.10. and synced using the
        GIT version control.

    3. Evaluate the space-time complexity of each major segment of the program, and the entire
        program, using big-O notation.

    3. Done, program comments. the total Big O of the program is O(n2).

    4. Explain the capability of your solution to scale and adapt to a growing number of packages.

    4. The algorithm is able to process and scale a greater number of packages; adding packages to
        the CSV file, and accounting for adding them to the trucks. for example: if increasing the
        number of packages to 200, the packages hash table will dynamically increase its list size.
        and as the delivery function is O(N2) time complexity, the increase in processing overhead is
        negligible. Thus, the solution is scalable and adaptable.

    5. Discuss why the software is efficient and easy to maintain.

    5. The software is reaching a 74.9 mileage for the trucks, most moving parts implement as
        classes and so they will be easy to maintain.

    6. Discuss the strengths and weaknesses of the self-adjusting data structures (e.g., the hash
        table).

C. Write an original program to deliver *all* the packages, meeting *all* requirements, using the attached supporting documents "Salt Lake City Downtown Map," "WGUPS Distance Table," and the "WGUPS Package File."
    1. Create an identifying comment within the first line of a file named "main.py" that includes your first name, last name, and student ID.
    2. Include comments in your code to explain the process and the flow of the program.

D. Identify a self-adjusting data structure, such as a hash table, that can 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 points you are storing.
D.

E. Develop a hash table, without using *any* additional libraries or classes, that has 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, en 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 (i.e., "at the hub," "en route," or "delivered"), including the delivery time

G. Provide an interface for the user to view the status and info (as listed in part F) 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 at a time between 8:35 a.m. and 9:25 a.m.
G.

1.

```
/Users/abdellahzakani/PycharmProjects/pythonProject/venv/bin/python /Users/abdellahzakani/PycharmProjects/Travelling-sales-problem/main.
The mileage for this specific route is:
86.89999999999999
Please enter a time during the travel of the route. Use the format, Hour:Minutes  9:00
To view the status of an individual package please type '1'. For all:   packages please type 'all'.  all
1, 195 W Oakland Ave, delivery deadline is: 10:30 AM,  Salt Lake City, 84115, 21 , Delivered at 8:39:00
2, 2530 S 500 E, delivery deadline is: EOD,  Salt Lake City, 84106, 44 , En route
3, 233 Canyon Rd, delivery deadline is: EOD,  Salt Lake City, 84103, 2 , En route
4, 380 W 2880 S, delivery deadline is: EOD,  Salt Lake City, 84115, 4 , Delivered at 8:42:40
5, 410 S State St, delivery deadline is: EOD,  Salt Lake City, 84111, 5 , En route
6, 3060 Lester St, delivery deadline is: 10:30 AM,  West Valley City, 84119, 88 , En route
7, 1330 2100 S, delivery deadline is: EOD,  Salt Lake City, 84106, 8 , Delivered at 8:29:40
8, 300 State St, delivery deadline is: EOD,  Salt Lake City, 84103, 9 , At Hub
9, 410 S State St, delivery deadline is: EOD,  Salt Lake City, 84111, 2 , En route
10, 600 E 900 South, delivery deadline is: EOD,  Salt Lake City, 84105, 1 , En route
11, 2600 Taylorsville Blvd, delivery deadline is: EOD,  Salt Lake City, 84118, 1 , En route
12, 3575 W Valley Central Station bus Loop, delivery deadline is: EOD,  West Valley City, 84119, 1 , En route
13, 2010 W 500 S, delivery deadline is: 10:30 AM,  Salt Lake City, 84104, 2 , At Hub
14, 4300 S 1300 E, delivery deadline is: 10:30 AM,  Millcreek, 84117, 88 , Delivered at 8:06:20
15, 4580 S 2300 E, delivery deadline is: 9:00 AM,  Holladay, 84117, 4 , Delivered at 8:13:00
16, 4580 S 2300 E, delivery deadline is: 10:30 AM,  Holladay, 84117, 88 , Delivered at 8:13:00
17, 3148 S 1100 W, delivery deadline is: EOD,  Salt Lake City, 84119, 2 , En route
18, 1488 4800 S, delivery deadline is: EOD,  Salt Lake City, 84123, 6 , En route
19, 177 W Price Ave, delivery deadline is: EOD,  Salt Lake City, 84115, 37 , Delivered at 8:49:40
20, 3595 Main St, delivery deadline is: 10:30 AM,  Salt Lake City, 84115, 37 , Delivered at 8:48:00
21, 3595 Main St, delivery deadline is: EOD,  Salt Lake City, 84115, 3 , En route
22, 6351 South 900 East, delivery deadline is: EOD,  Murray, 84121, 2 , En route
23, 5100 South 2700 West, delivery deadline is: EOD,  Salt Lake City, 84118, 5 , En route
24, 5025 State St, delivery deadline is: EOD,  Murray, 84107, 7 , En route
25, 5383 South 900 East #104, delivery deadline is: 10:30 AM,  Salt Lake City, 84117, 7 , En route
26, 5383 South 900 East #104, delivery deadline is: EOD,  Salt Lake City, 84117, 25 , En route
27, 1060 Dalton Ave S, delivery deadline is: EOD,  Salt Lake City, 84104, 5 , En route
28, 2835 Main St, delivery deadline is: EOD,  Salt Lake City, 84115, 7 , En route
29, 1330 2100 S, delivery deadline is: 10:30 AM,  Salt Lake City, 84106, 2 , Delivered at 8:29:40
30, 300 State St, delivery deadline is: 10:30 AM,  Salt Lake City, 84103, 1 , At Hub
31, 3365 S 900 W, delivery deadline is: 10:30 AM,  Salt Lake City, 84119, 1 , Delivered at 8:58:40
32, 3365 S 900 W, delivery deadline is: EOD,  Salt Lake City, 84119, 1 , En route
33, 2530 S 500 E, delivery deadline is: EOD,  Salt Lake City, 84106, 1 , En route
34, 4580 S 2300 E, delivery deadline is: 10:30 AM,  Holladay, 84117, 2 , Delivered at 8:13:00
35, 1060 Dalton Ave S, delivery deadline is: EOD,  Salt Lake City, 84104, 88 , En route
```

2. Provide screenshots to show the status of *all* packages at a time between 9:35 a.m. and 10:25 a.m.

3. Provide screenshots to show the status of *all* packages at a time between 12:03 p.m. and 1:12 p.m.

H. Provide a screenshot or screenshots showing the successful completion of the code, free from runtime errors or warnings, that includes the total mileage traveled by *all* trucks.

I. Justify the core algorithm you identified in part A and used in the solution by doing the following:
1. Describe *at least* **two** strengths of the algorithm used in the solution.
1. The two strengths of the nearest neighbor algorithm is a simplicity to implement, only 3 lines of code; and speed the time complexity is O(N).
2. Verify that the algorithm used in the solution meets *all* requirements in the scenario.
2. The algorithm meets all requirements.
3. Identify **two** other named algorithms, different from the algorithm implemented in the solution that would meet the requirements in the scenario.
3. Two other algorithms can be Brute force and randomized algorithms
   a. Describe how *each* algorithm identified in part I3 is different from the algorithm used in the solution.
   a. Brute force can be used to run all possible combinations of location deliveries to find the most optimized solution; a contrast difference where the nearest neighbor algorithm checks for the next closest package instead.
   Randomized can be used to deliver randomly and test for completion of requirements. different from the nearest neighbor algorithm.
J. Describe what you would do differently, other than the two algorithms identified in I3, if you did this project again.

J. I would use graphing tools to solve the requirements and possibly integrate the location with a Google Map API.

K. Justify the data structure you identified in part D by doing the following:
1. Verify that the data structure used in the solution meets *all* requirements in the scenario.
    a. Explain how the time needed to complete the look-up function is affected by changes in the number of packages to be delivered.

K.
1.
    a. The time needed to execute a lookup function is not affected by the number of packages as we used a hash map and the look function is O (1).
    b. Explain how the data structure space usage is affected by changes in the number of packages to be delivered.
    b. The space used is affected linearly by the number of packages.
    c. Describe how changes to the number of trucks or the number of cities would affect the look-up time and the space usage of the data structure.
    c. A hash map has used the change of cities or the number of trucks will affect lookup time, but it would affect lookup space if more data is added.
2. Identify **two** other data structures that could meet the same requirements in the scenario.
    a. Describe how *each* data structure identified in part K2 is different from the data structure used in the solution.
2. The data structures that can be used are lists or dictionaries.
    A. Dictionaries are different than Hashtable in the way they are implemented, the dictionaries are implemented as an array and the hashtable is implemented as a linked list. hash tables are also faster than dictionaries, as dictionaries' lookup compares each element; hash tables do not.
the same difference in implementation between lists and hash tables; items are accessed via their index in lists, and accessed via hash values/keys in hash tables.