C950 Program Overview

Core problem and assumptions:

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 route has three trucks, two drivers, and an average of 40 packages to deliver each day. The goal of the project is to create an algorithm to efficiently deliver all of the packages in a timely manner.

The following assumptions were taken into account during the development of the program:

- 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
- 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.

Section A: self-adjusting algorithm

The algorithm used in this program to deliver packages was the nearest neighbor algorithm. This function takes the data given and decides which point will be closest after its calculations and sets it as the next target destination. Once the new destination becomes the main destination, the algorithm will repeat the process and do a calculation on the next point.

Section B:

- The general nearest neighbor algorithm works in the works as follows:

  - Initialize all vertices as unvisited.

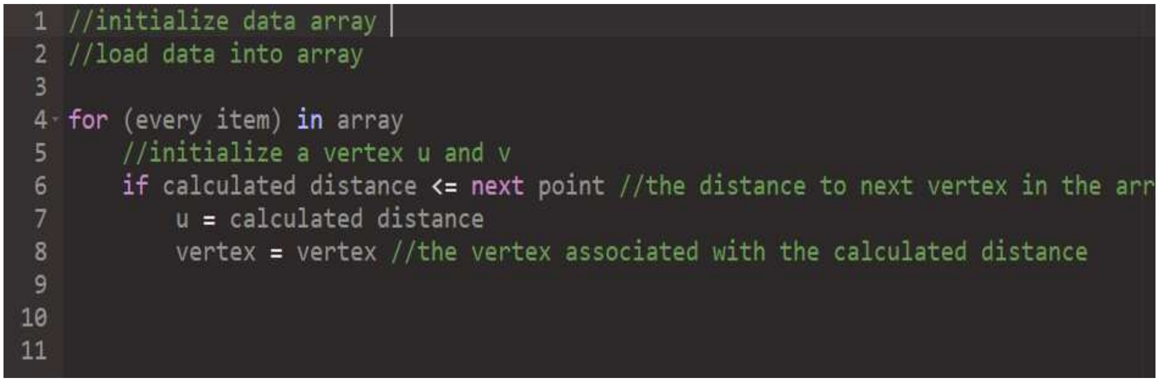
  - Select any vertex, set it as the current vertex and set as visited.

  - Find the shortest edge connecting the current vertex **u** and an unvisited new vertex **v**.

  - Set **v** as the new current vertex **u and m**ark **v** as visited.

  - End once every point has been visited

  (Pseudocode for general nearest neighbor algorithm)



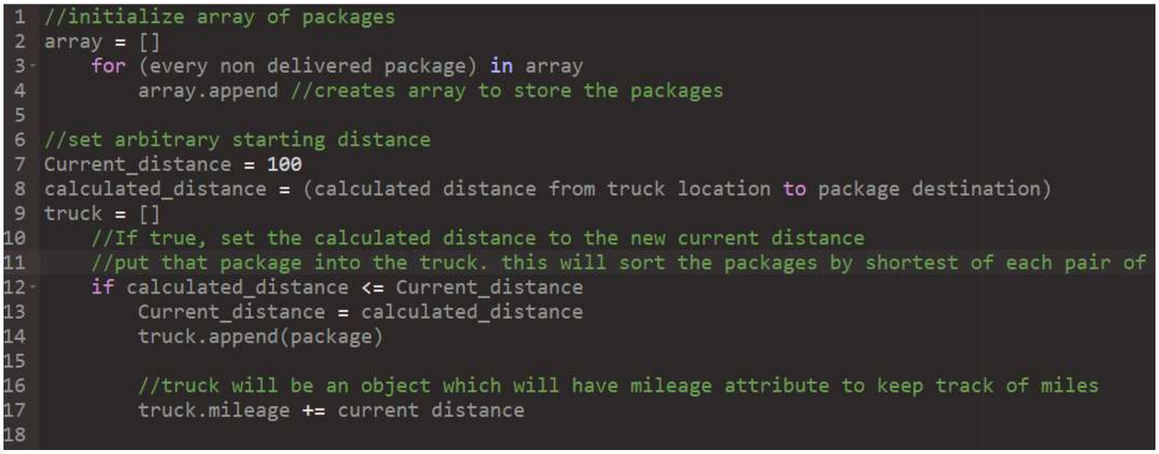
```
//initialize data array
//load data into array

for (every item) in array
    //initialize a vertex u and v
    if calculated distance <= next point //the distance to next vertex in the arr
        u = calculated distance
        vertex = vertex //the vertex associated with the calculated distance



```

(Pseudocode for nearest neighbor algorithm in this application)



```
//initialize array of packages
array = []
    for (every non delivered package) in array
        array.append //creates array to store the packages

//set arbitrary starting distance
Current_distance = 100
calculated_distance = (calculated distance from truck location to package destination)
truck = []
    //If true, set the calculated distance to the new current distance
    //put that package into the truck. this will sort the packages by shortest of each pair of
    if calculated_distance <= Current_distance
        Current_distance = calculated_distance
        truck.append(package)

        //truck will be an object which will have mileage attribute to keep track of miles
        truck.mileage += current distance

```

For this particular example, since for loop only iterates through each item of the array once, the time complexity of the loop is O(n). The calculated distance will be a function to compare the distance between one point and all other points which will also be O(n). These time and space complexities will give this algorithm a value of O(n^2).

- The environment used to develop this program was the latest version of PyCharm 2022.3 Community edition with python 3.9 (system default) in windows 10 Pro on desktop PC. Ryzen 5 processor, 32gb ram with windows running on m.2 nvme SSD.

- Time-Space complexity analysis of different parts of the program. The time complexity and space complexity will be the same in this case (In depth notes on complexity in the comments of the code)

(HashMap.py)

Method	**Time and Space Complexity**
__init__	O(n)
insert	O(n)
search	O(n)
hash_remove	O(1)
Total complexity	O(n)+O(n)+O(n)+O(1) = 3*O(n)+O(1)
Total	O(n)

(Package.py)

Method	Time and Space Complexity
packageStatus	O(1)
Total	O(1)

(Main.py)

Method	Time and Space Complexity
calculateDistance	O(1)
getAddress	O(n)
startData	O(n)
deliveryRoute	O(n^2)
Class Main:	O(n)
Total complexity	O(n^2)+3*O(n)+O(1)
Total	O(n^2)

Total overall time and space complexity = O(n^2) + O(n) + O(1) = **O(n^2) (polynomial complexity)**

- Scalability and adaptability are necessary for programs like these as postal services get many more packages than in the given scenario and need to be able to efficiently handle them all. By using a chaining hash table in this application, it will be able to dynamically adjust to any number of packages put in the system. This data can be easily loaded from any .csv file as long as the data inputs are in the correct format. The algorithm will also dynamically grow automatically as packages are added as it will always iterate through the entire list of packages no matter how many there are. It is initialized with an empty array to be loaded with packages from the truck so it can take as many packages as necessary.

- To achieve a good level of efficiency, the application was developed using an object-oriented approach which will items such as packages and trucks easier to adjust and manipulate in future iterations. For maintainability, each class and method have plenty of well-documented comments describing how everything works for any other developers taking on the project.

- The benefit of using a chaining hash table is that it ensures that there will always be enough room for an ‘n’ number of packages while also making sure there aren’t any collisions. This means that as the number of packages increases, the list size will increase as well to accommodate them. A chaining hash is an array of linked lists. The only real weakness of it, however, is as the table grows it increases the overall time complexity of insertion and deletion into the table. This can be handled by increasing the size of the hash at initialization so as the packet numbers grow it will have less work to do.

Section D:

The data structure used in this program was a chaining hash table with insert, search, and remove capabilities using lists and arrays.

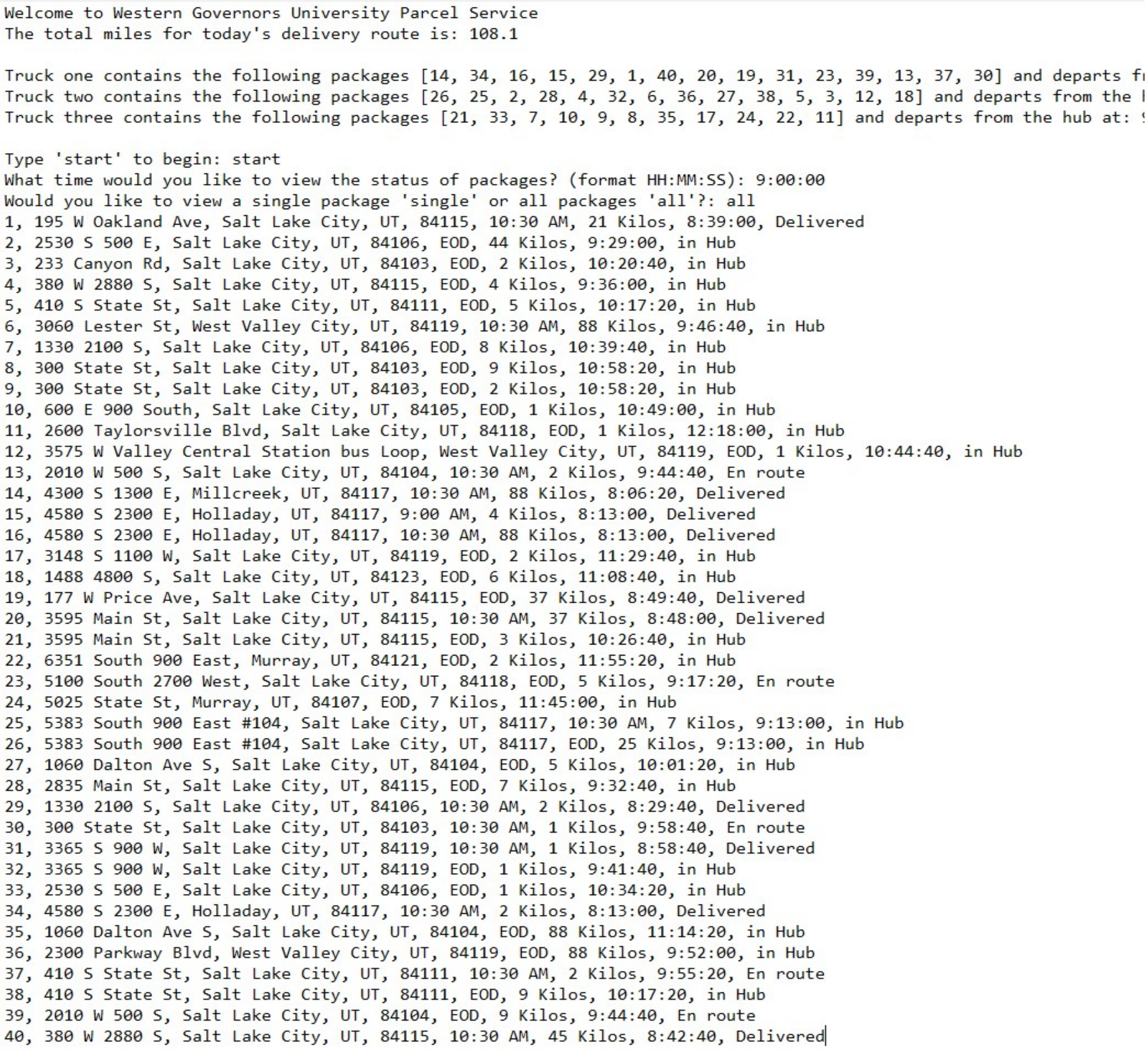
1) In a chaining hash table, data is stored in an array of buckets using a hashing function to determine the index at which each piece of data should be stored. In this case, that would be the packages. This function maps the packages to be stored to specific indices in the array, creating a relationship between the data and its storage location in the hash table.

When packages are inserted into the hash table, the hashing function is used to determine the index at which the data should be stored. This allows for efficient insertion of data, as the index can be quickly calculated and the data can be placed in the appropriate bucket. When data is retrieved from a hash table with the search function, the hashing function is again used to determine the index at which the data is stored, allowing for efficient retrieval of the data.

The relationship between the data points and their storage locations in a hash table is determined by the hashing function, which maps the data to specific indices in the array. This allows for efficient insertion and retrieval of data in the hash table.

Section G:

- Time between 8:35 a.m. and 9:25 a.m. (could not fit full query in screenshot view. Copied to notepad from interface to get every line in the shot)



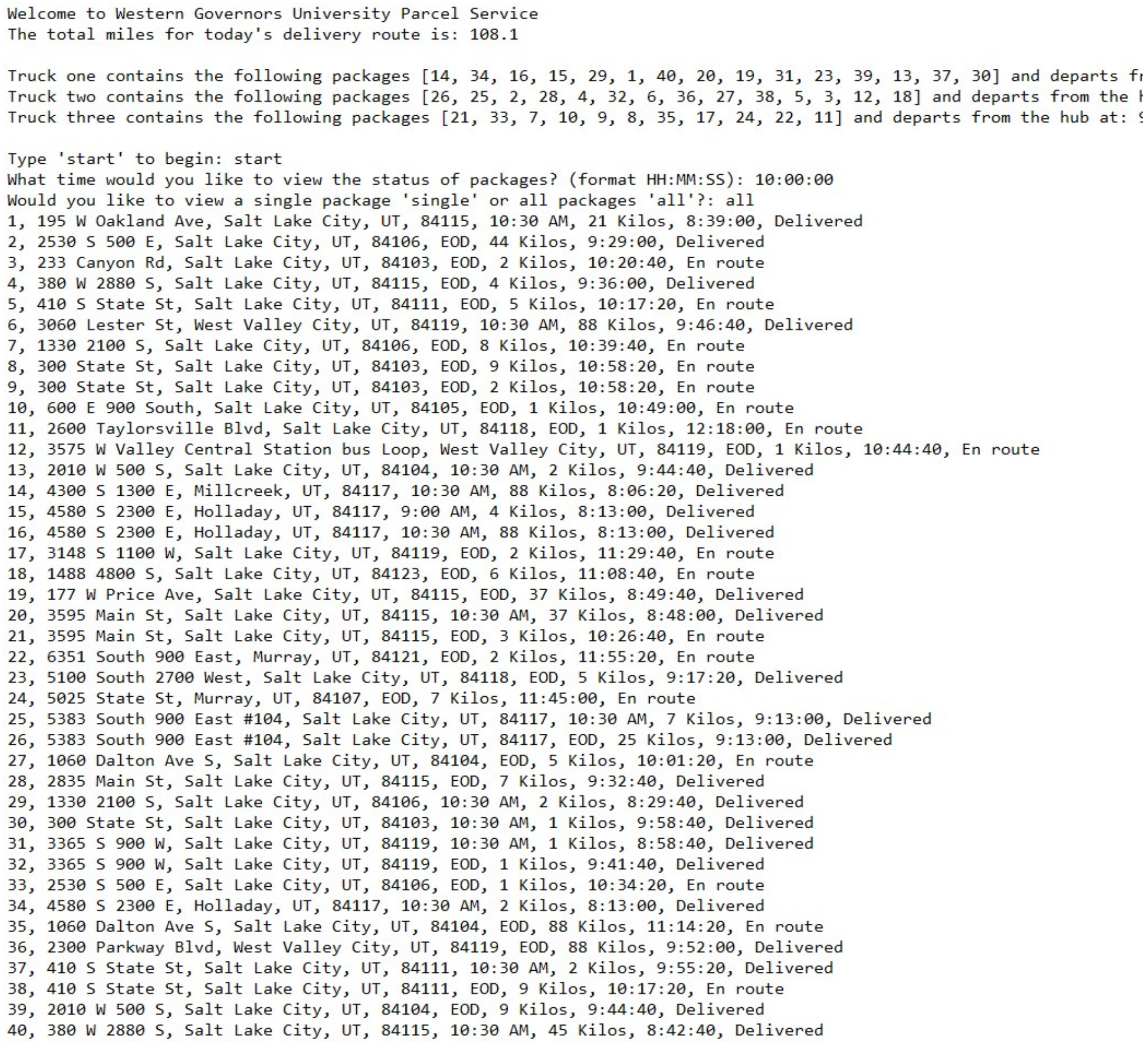
```
Welcome to Western Governors University Parcel Service
The total miles for today's delivery route is: 108.1

Truck one contains the following packages [14, 34, 16, 15, 29, 1, 40, 20, 19, 31, 23, 39, 13, 37, 30] and departs f
Truck two contains the following packages [26, 25, 2, 28, 4, 32, 6, 36, 27, 38, 5, 3, 12, 18] and departs from the
Truck three contains the following packages [21, 33, 7, 10, 9, 8, 35, 17, 24, 22, 11] and departs from the hub at:

Type 'start' to begin: start
What time would you like to view the status of packages? (format HH:MM:SS): 9:00:00
Would you like to view a single package 'single' or all packages 'all'?: all
1, 195 W Oakland Ave, Salt Lake City, UT, 84115, 10:30 AM, 21 Kilos, 8:39:00, Delivered
2, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 44 Kilos, 9:29:00, in Hub
3, 233 Canyon Rd, Salt Lake City, UT, 84103, EOD, 2 Kilos, 10:20:40, in Hub
4, 380 W 2880 S, Salt Lake City, UT, 84115, EOD, 4 Kilos, 9:36:00, in Hub
5, 410 S State St, Salt Lake City, UT, 84111, EOD, 5 Kilos, 10:17:20, in Hub
6, 3060 Lester St, West Valley City, UT, 84119, 10:30 AM, 88 Kilos, 9:46:40, in Hub
7, 1330 2100 S, Salt Lake City, UT, 84106, EOD, 8 Kilos, 10:39:40, in Hub
8, 300 State St, Salt Lake City, UT, 84103, EOD, 9 Kilos, 10:58:20, in Hub
9, 300 State St, Salt Lake City, UT, 84103, EOD, 2 Kilos, 10:58:20, in Hub
10, 600 E 900 South, Salt Lake City, UT, 84105, EOD, 1 Kilos, 10:49:00, in Hub
11, 2600 Taylorsville Blvd, Salt Lake City, UT, 84118, EOD, 1 Kilos, 12:18:00, in Hub
12, 3575 W Valley Central Station bus Loop, West Valley City, UT, 84119, EOD, 1 Kilos, 10:44:40, in Hub
13, 2010 W 500 S, Salt Lake City, UT, 84104, 10:30 AM, 2 Kilos, 9:44:40, En route
14, 4300 S 1300 E, Millcreek, UT, 84117, 10:30 AM, 88 Kilos, 8:06:20, Delivered
15, 4580 S 2300 E, Holladay, UT, 84117, 9:00 AM, 4 Kilos, 8:13:00, Delivered
16, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 88 Kilos, 8:13:00, Delivered
17, 3148 S 1100 W, Salt Lake City, UT, 84119, EOD, 2 Kilos, 11:29:40, in Hub
18, 1488 4800 S, Salt Lake City, UT, 84123, EOD, 6 Kilos, 11:08:40, in Hub
19, 177 W Price Ave, Salt Lake City, UT, 84115, EOD, 37 Kilos, 8:49:40, Delivered
20, 3595 Main St, Salt Lake City, UT, 84115, 10:30 AM, 37 Kilos, 8:48:00, Delivered
21, 3595 Main St, Salt Lake City, UT, 84115, EOD, 3 Kilos, 10:26:40, in Hub
22, 6351 South 900 East, Murray, UT, 84121, EOD, 2 Kilos, 11:55:20, in Hub
23, 5100 South 2700 West, Salt Lake City, UT, 84118, EOD, 5 Kilos, 9:17:20, En route
24, 5025 State St, Murray, UT, 84107, EOD, 7 Kilos, 11:45:00, in Hub
25, 5383 South 900 East #104, Salt Lake City, UT, 84117, 10:30 AM, 7 Kilos, 9:13:00, in Hub
26, 5383 South 900 East #104, Salt Lake City, UT, 84117, EOD, 25 Kilos, 9:13:00, in Hub
27, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 5 Kilos, 10:01:20, in Hub
28, 2835 Main St, Salt Lake City, UT, 84115, EOD, 7 Kilos, 9:32:40, in Hub
29, 1330 2100 S, Salt Lake City, UT, 84106, 10:30 AM, 2 Kilos, 8:29:40, Delivered
30, 300 State St, Salt Lake City, UT, 84103, 10:30 AM, 1 Kilos, 9:58:40, En route
31, 3365 S 900 W, Salt Lake City, UT, 84119, 10:30 AM, 1 Kilos, 8:58:40, Delivered
32, 3365 S 900 W, Salt Lake City, UT, 84119, EOD, 1 Kilos, 9:41:40, in Hub
33, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 1 Kilos, 10:34:20, in Hub
34, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 2 Kilos, 8:13:00, Delivered
35, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 88 Kilos, 11:14:20, in Hub
36, 2300 Parkway Blvd, West Valley City, UT, 84119, EOD, 88 Kilos, 9:52:00, in Hub
37, 410 S State St, Salt Lake City, UT, 84111, 10:30 AM, 2 Kilos, 9:55:20, En route
38, 410 S State St, Salt Lake City, UT, 84111, EOD, 9 Kilos, 10:17:20, in Hub
39, 2010 W 500 S, Salt Lake City, UT, 84104, EOD, 9 Kilos, 9:44:40, En route
40, 380 W 2880 S, Salt Lake City, UT, 84115, 10:30 AM, 45 Kilos, 8:42:40, Delivered
```

- 9:35 a.m. and 10:25 a.m. (package 9 will not update address until exactly 10:20

am)

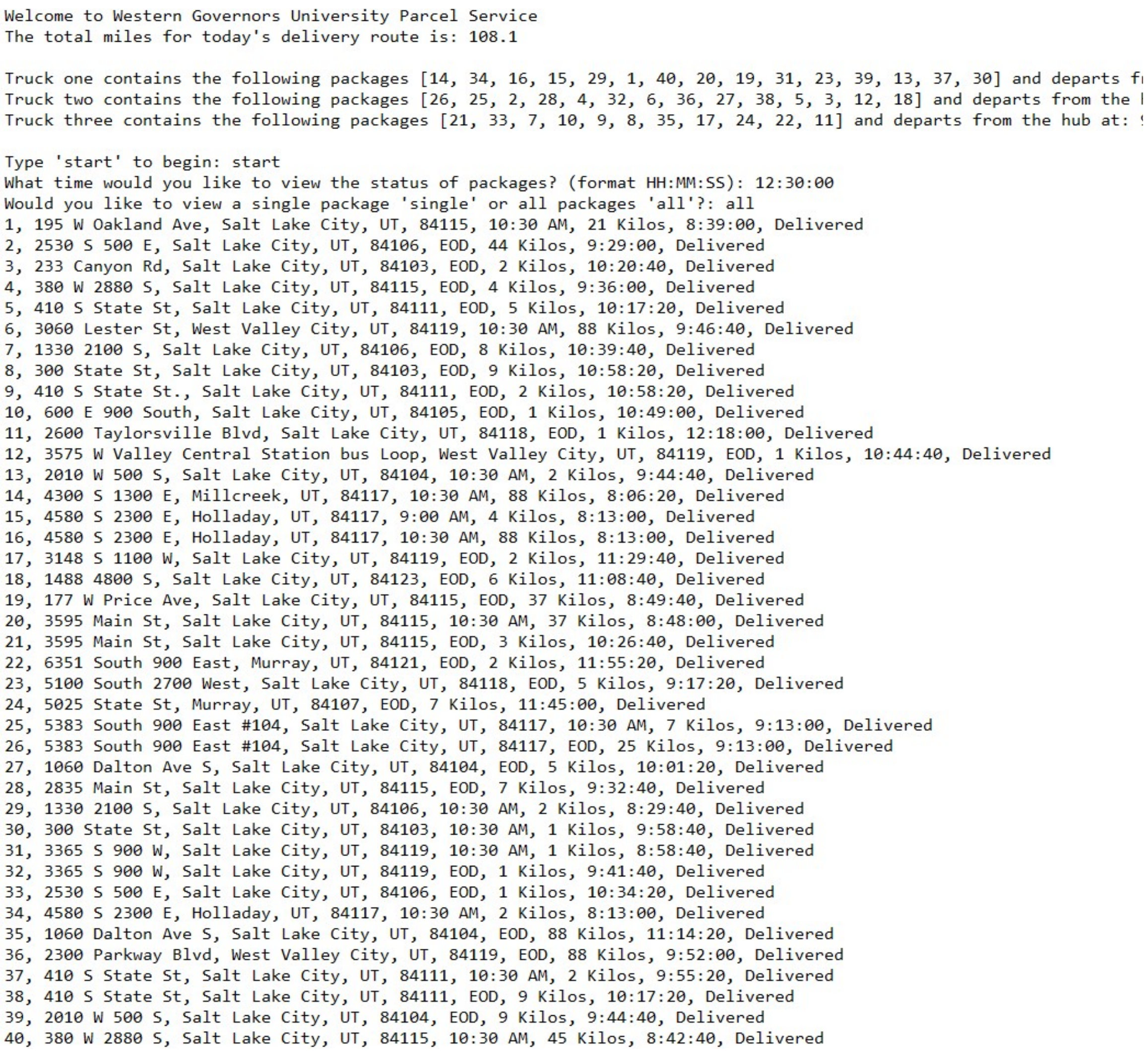


```
Welcome to Western Governors University Parcel Service
The total miles for today's delivery route is: 108.1

Truck one contains the following packages [14, 34, 16, 15, 29, 1, 40, 20, 19, 31, 23, 39, 13, 37, 30] and departs fr
Truck two contains the following packages [26, 25, 2, 28, 4, 32, 6, 36, 27, 38, 5, 3, 12, 18] and departs from the h
Truck three contains the following packages [21, 33, 7, 10, 9, 8, 35, 17, 24, 22, 11] and departs from the hub at: 9

Type 'start' to begin: start
What time would you like to view the status of packages? (format HH:MM:SS): 10:00:00
Would you like to view a single package 'single' or all packages 'all'?: all
1, 195 W Oakland Ave, Salt Lake City, UT, 84115, 10:30 AM, 21 Kilos, 8:39:00, Delivered
2, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 44 Kilos, 9:29:00, Delivered
3, 233 Canyon Rd, Salt Lake City, UT, 84103, EOD, 2 Kilos, 10:20:40, En route
4, 380 W 2880 S, Salt Lake City, UT, 84115, EOD, 4 Kilos, 9:36:00, Delivered
5, 410 S State St, Salt Lake City, UT, 84111, EOD, 5 Kilos, 10:17:20, En route
6, 3060 Lester St, West Valley City, UT, 84119, 10:30 AM, 88 Kilos, 9:46:40, Delivered
7, 1330 2100 S, Salt Lake City, UT, 84106, EOD, 8 Kilos, 10:39:40, En route
8, 300 State St, Salt Lake City, UT, 84103, EOD, 9 Kilos, 10:58:20, En route
9, 300 State St, Salt Lake City, UT, 84103, EOD, 2 Kilos, 10:58:20, En route
10, 600 E 900 South, Salt Lake City, UT, 84105, EOD, 1 Kilos, 10:49:00, En route
11, 2600 Taylorsville Blvd, Salt Lake City, UT, 84118, EOD, 1 Kilos, 12:18:00, En route
12, 3575 W Valley Central Station bus Loop, West Valley City, UT, 84119, EOD, 1 Kilos, 10:44:40, En route
13, 2010 W 500 S, Salt Lake City, UT, 84104, 10:30 AM, 2 Kilos, 9:44:40, Delivered
14, 4300 S 1300 E, Millcreek, UT, 84117, 10:30 AM, 88 Kilos, 8:06:20, Delivered
15, 4580 S 2300 E, Holladay, UT, 84117, 9:00 AM, 4 Kilos, 8:13:00, Delivered
16, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 88 Kilos, 8:13:00, Delivered
17, 3148 S 1100 W, Salt Lake City, UT, 84119, EOD, 2 Kilos, 11:29:40, En route
18, 1488 4800 S, Salt Lake City, UT, 84123, EOD, 6 Kilos, 11:08:40, En route
19, 177 W Price Ave, Salt Lake City, UT, 84115, EOD, 37 Kilos, 8:49:40, Delivered
20, 3595 Main St, Salt Lake City, UT, 84115, 10:30 AM, 37 Kilos, 8:48:00, Delivered
21, 3595 Main St, Salt Lake City, UT, 84115, EOD, 3 Kilos, 10:26:40, En route
22, 6351 South 900 East, Murray, UT, 84121, EOD, 2 Kilos, 11:55:20, En route
23, 5100 South 2700 West, Salt Lake City, UT, 84118, EOD, 5 Kilos, 9:17:20, Delivered
24, 5025 State St, Murray, UT, 84107, EOD, 7 Kilos, 11:45:00, En route
25, 5383 South 900 East #104, Salt Lake City, UT, 84117, 10:30 AM, 7 Kilos, 9:13:00, Delivered
26, 5383 South 900 East #104, Salt Lake City, UT, 84117, EOD, 25 Kilos, 9:13:00, Delivered
27, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 5 Kilos, 10:01:20, En route
28, 2835 Main St, Salt Lake City, UT, 84115, EOD, 7 Kilos, 9:32:40, Delivered
29, 1330 2100 S, Salt Lake City, UT, 84106, 10:30 AM, 2 Kilos, 8:29:40, Delivered
30, 300 State St, Salt Lake City, UT, 84103, 10:30 AM, 1 Kilos, 9:58:40, Delivered
31, 3365 S 900 W, Salt Lake City, UT, 84119, 10:30 AM, 1 Kilos, 8:58:40, Delivered
32, 3365 S 900 W, Salt Lake City, UT, 84119, EOD, 1 Kilos, 9:41:40, Delivered
33, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 1 Kilos, 10:34:20, En route
34, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 2 Kilos, 8:13:00, Delivered
35, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 88 Kilos, 11:14:20, En route
36, 2300 Parkway Blvd, West Valley City, UT, 84119, EOD, 88 Kilos, 9:52:00, Delivered
37, 410 S State St, Salt Lake City, UT, 84111, 10:30 AM, 2 Kilos, 9:55:20, Delivered
38, 410 S State St, Salt Lake City, UT, 84111, EOD, 9 Kilos, 10:17:20, En route
39, 2010 W 500 S, Salt Lake City, UT, 84104, EOD, 9 Kilos, 9:44:40, Delivered
40, 380 W 2880 S, Salt Lake City, UT, 84115, 10:30 AM, 45 Kilos, 8:42:40, Delivered
```

- time between 12:03 p.m. and 1:12 p.m. (package 9 will update address directly at 10:20 am)



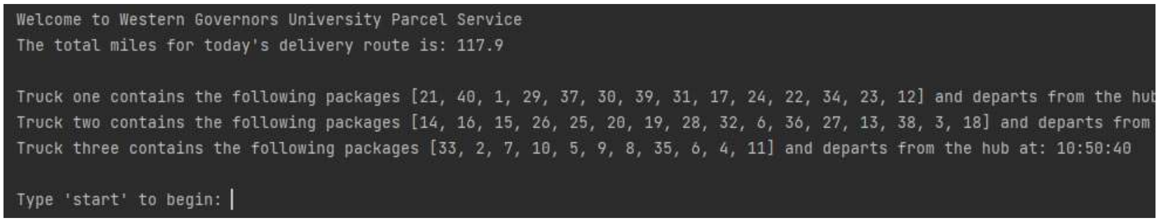
```
Welcome to Western Governors University Parcel Service
The total miles for today's delivery route is: 108.1

Truck one contains the following packages [14, 34, 16, 15, 29, 1, 40, 20, 19, 31, 23, 39, 13, 37, 30] and departs fr
Truck two contains the following packages [26, 25, 2, 28, 4, 32, 6, 36, 27, 38, 5, 3, 12, 18] and departs from the
Truck three contains the following packages [21, 33, 7, 10, 9, 8, 35, 17, 24, 22, 11] and departs from the hub at:

Type 'start' to begin: start
What time would you like to view the status of packages? (format HH:MM:SS): 12:30:00
Would you like to view a single package 'single' or all packages 'all'?: all
1, 195 W Oakland Ave, Salt Lake City, UT, 84115, 10:30 AM, 21 Kilos, 8:39:00, Delivered
2, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 44 Kilos, 9:29:00, Delivered
3, 233 Canyon Rd, Salt Lake City, UT, 84103, EOD, 2 Kilos, 10:20:40, Delivered
4, 380 W 2880 S, Salt Lake City, UT, 84115, EOD, 4 Kilos, 9:36:00, Delivered
5, 410 S State St, Salt Lake City, UT, 84111, EOD, 5 Kilos, 10:17:20, Delivered
6, 3060 Lester St, West Valley City, UT, 84119, 10:30 AM, 88 Kilos, 9:46:40, Delivered
7, 1330 2100 S, Salt Lake City, UT, 84106, EOD, 8 Kilos, 10:39:40, Delivered
8, 300 State St, Salt Lake City, UT, 84103, EOD, 9 Kilos, 10:58:20, Delivered
9, 410 S State St., Salt Lake City, UT, 84111, EOD, 2 Kilos, 10:58:20, Delivered
10, 600 E 900 South, Salt Lake City, UT, 84105, EOD, 1 Kilos, 10:49:00, Delivered
11, 2600 Taylorsville Blvd, Salt Lake City, UT, 84118, EOD, 1 Kilos, 12:18:00, Delivered
12, 3575 W Valley Central Station bus Loop, West Valley City, UT, 84119, EOD, 1 Kilos, 10:44:40, Delivered
13, 2010 W 500 S, Salt Lake City, UT, 84104, 10:30 AM, 2 Kilos, 9:44:40, Delivered
14, 4300 S 1300 E, Millcreek, UT, 84117, 10:30 AM, 88 Kilos, 8:06:20, Delivered
15, 4580 S 2300 E, Holladay, UT, 84117, 9:00 AM, 4 Kilos, 8:13:00, Delivered
16, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 88 Kilos, 8:13:00, Delivered
17, 3148 S 1100 W, Salt Lake City, UT, 84119, EOD, 2 Kilos, 11:29:40, Delivered
18, 1488 4800 S, Salt Lake City, UT, 84123, EOD, 6 Kilos, 11:08:40, Delivered
19, 177 W Price Ave, Salt Lake City, UT, 84115, EOD, 37 Kilos, 8:49:40, Delivered
20, 3595 Main St, Salt Lake City, UT, 84115, 10:30 AM, 37 Kilos, 8:48:00, Delivered
21, 3595 Main St, Salt Lake City, UT, 84115, EOD, 3 Kilos, 10:26:40, Delivered
22, 6351 South 900 East, Murray, UT, 84121, EOD, 2 Kilos, 11:55:20, Delivered
23, 5100 South 2700 West, Salt Lake City, UT, 84118, EOD, 5 Kilos, 9:17:20, Delivered
24, 5025 State St, Murray, UT, 84107, EOD, 7 Kilos, 11:45:00, Delivered
25, 5383 South 900 East #104, Salt Lake City, UT, 84117, 10:30 AM, 7 Kilos, 9:13:00, Delivered
26, 5383 South 900 East #104, Salt Lake City, UT, 84117, EOD, 25 Kilos, 9:13:00, Delivered
27, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 5 Kilos, 10:01:20, Delivered
28, 2835 Main St, Salt Lake City, UT, 84115, EOD, 7 Kilos, 9:32:40, Delivered
29, 1330 2100 S, Salt Lake City, UT, 84106, 10:30 AM, 2 Kilos, 8:29:40, Delivered
30, 300 State St, Salt Lake City, UT, 84103, 10:30 AM, 1 Kilos, 9:58:40, Delivered
31, 3365 S 900 W, Salt Lake City, UT, 84119, 10:30 AM, 1 Kilos, 8:58:40, Delivered
32, 3365 S 900 W, Salt Lake City, UT, 84119, EOD, 1 Kilos, 9:41:40, Delivered
33, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 1 Kilos, 10:34:20, Delivered
34, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 2 Kilos, 8:13:00, Delivered
35, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 88 Kilos, 11:14:20, Delivered
36, 2300 Parkway Blvd, West Valley City, UT, 84119, EOD, 88 Kilos, 9:52:00, Delivered
37, 410 S State St, Salt Lake City, UT, 84111, 10:30 AM, 2 Kilos, 9:55:20, Delivered
38, 410 S State St, Salt Lake City, UT, 84111, EOD, 9 Kilos, 10:17:20, Delivered
39, 2010 W 500 S, Salt Lake City, UT, 84104, EOD, 9 Kilos, 9:44:40, Delivered
40, 380 W 2880 S, Salt Lake City, UT, 84115, 10:30 AM, 45 Kilos, 8:42:40, Delivered
```

Section H: Completion of code

(Total mileage for the route. All three trucks combined)

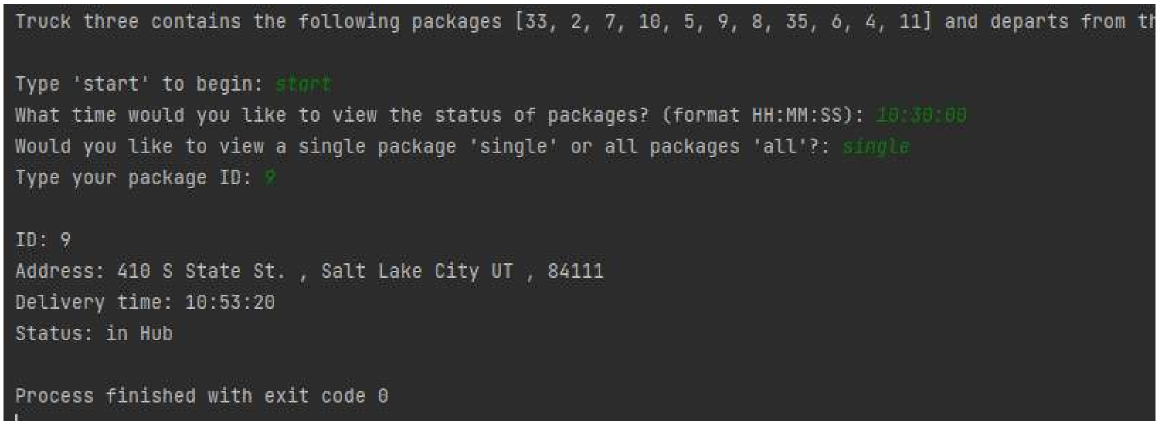


```
Welcome to Western Governors University Parcel Service
The total miles for today's delivery route is: 117.9

Truck one contains the following packages [21, 40, 1, 29, 37, 30, 39, 31, 17, 24, 22, 34, 23, 12] and departs from the hub
Truck two contains the following packages [14, 16, 15, 26, 25, 20, 19, 28, 32, 6, 36, 27, 13, 38, 3, 18] and departs from
Truck three contains the following packages [33, 2, 7, 10, 5, 9, 8, 35, 6, 4, 11] and departs from the hub at: 10:50:40

Type 'start' to begin: |
```

(Searching for a single package)



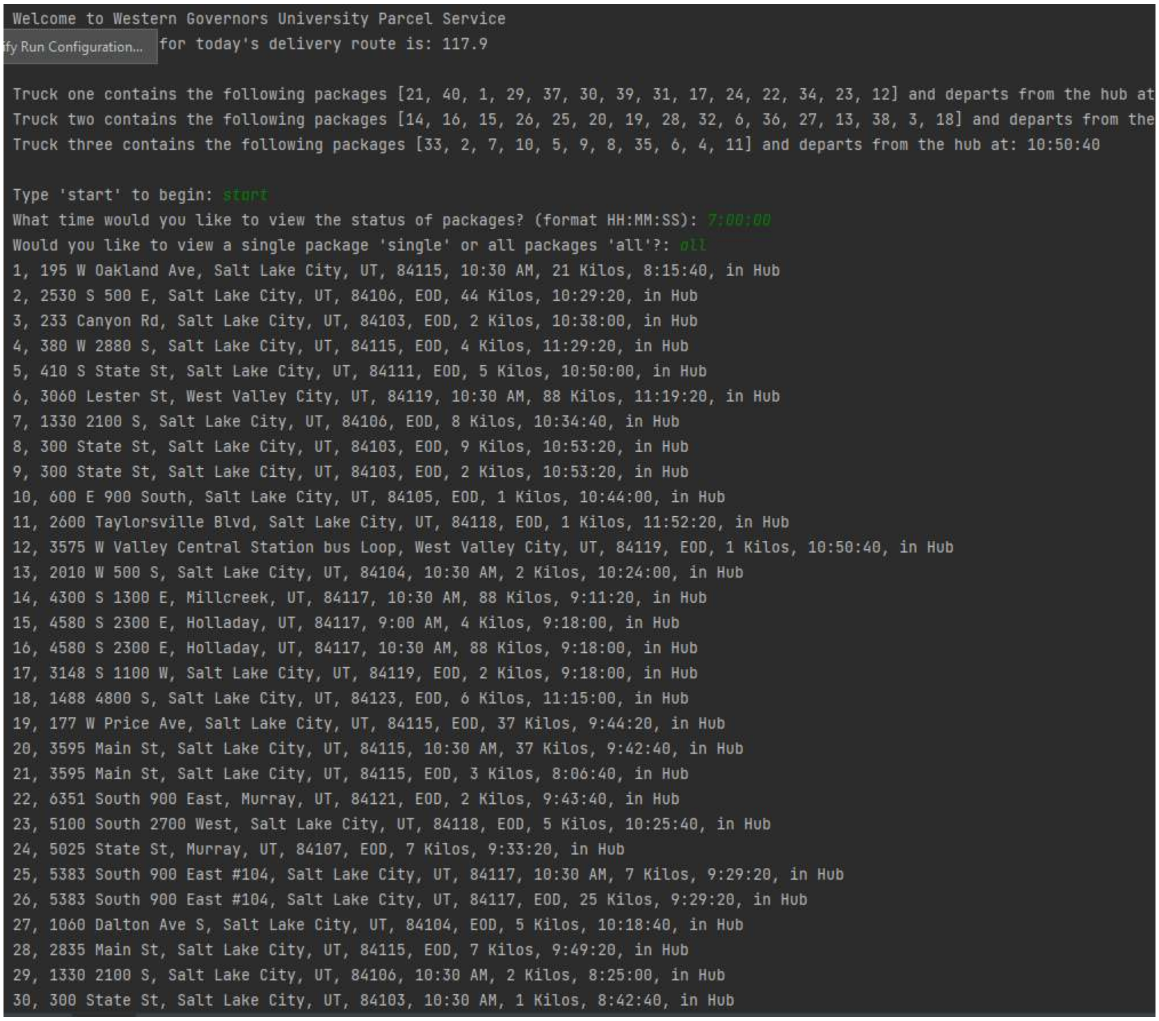
```
Truck three contains the following packages [33, 2, 7, 10, 5, 9, 8, 35, 6, 4, 11] and departs from th

Type 'start' to begin: start
What time would you like to view the status of packages? (format HH:MM:SS): 10:30:00
Would you like to view a single package 'single' or all packages 'all'?: single
Type your package ID: 9

ID: 9
Address: 410 S State St. , Salt Lake City UT , 84111
Delivery time: 10:53:20
Status: in Hub

Process finished with exit code 0
```

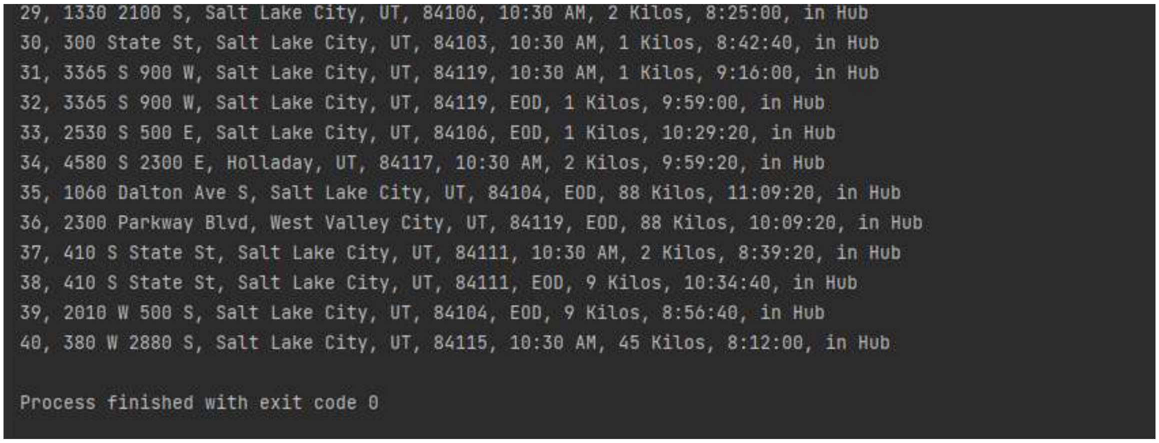
(Searching all packages. Searched before any trucks depart to show they are all in the hub.)



```
Welcome to Western Governors University Parcel Service
 for today's delivery route is: 117.9

Truck one contains the following packages [21, 40, 1, 29, 37, 30, 39, 31, 17, 24, 22, 34, 23, 12] and departs from the hub at
Truck two contains the following packages [14, 16, 15, 26, 25, 20, 19, 28, 32, 6, 36, 27, 13, 38, 3, 18] and departs from the
Truck three contains the following packages [33, 2, 7, 10, 5, 9, 8, 35, 6, 4, 11] and departs from the hub at: 10:50:40

Type 'start' to begin: start
What time would you like to view the status of packages? (format HH:MM:SS): 7:00:00
Would you like to view a single package 'single' or all packages 'all'?: all
1, 195 W Oakland Ave, Salt Lake City, UT, 84115, 10:30 AM, 21 Kilos, 8:15:40, in Hub
2, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 44 Kilos, 10:29:20, in Hub
3, 233 Canyon Rd, Salt Lake City, UT, 84103, EOD, 2 Kilos, 10:38:00, in Hub
4, 380 W 2880 S, Salt Lake City, UT, 84115, EOD, 4 Kilos, 11:29:20, in Hub
5, 410 S State St, Salt Lake City, UT, 84111, EOD, 5 Kilos, 10:50:00, in Hub
6, 3060 Lester St, West Valley City, UT, 84119, 10:30 AM, 88 Kilos, 11:19:20, in Hub
7, 1330 2100 S, Salt Lake City, UT, 84106, EOD, 8 Kilos, 10:34:40, in Hub
8, 300 State St, Salt Lake City, UT, 84103, EOD, 9 Kilos, 10:53:20, in Hub
9, 300 State St, Salt Lake City, UT, 84103, EOD, 2 Kilos, 10:53:20, in Hub
10, 600 E 900 South, Salt Lake City, UT, 84105, EOD, 1 Kilos, 10:44:00, in Hub
11, 2600 Taylorsville Blvd, Salt Lake City, UT, 84118, EOD, 1 Kilos, 11:52:20, in Hub
12, 3575 W Valley Central Station bus Loop, West Valley City, UT, 84119, EOD, 1 Kilos, 10:50:40, in Hub
13, 2010 W 500 S, Salt Lake City, UT, 84104, 10:30 AM, 2 Kilos, 10:24:00, in Hub
14, 4300 S 1300 E, Millcreek, UT, 84117, 10:30 AM, 88 Kilos, 9:11:20, in Hub
15, 4580 S 2300 E, Holladay, UT, 84117, 9:00 AM, 4 Kilos, 9:18:00, in Hub
16, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 88 Kilos, 9:18:00, in Hub
17, 3148 S 1100 W, Salt Lake City, UT, 84119, EOD, 2 Kilos, 9:18:00, in Hub
18, 1488 4800 S, Salt Lake City, UT, 84123, EOD, 6 Kilos, 11:15:00, in Hub
19, 177 W Price Ave, Salt Lake City, UT, 84115, EOD, 37 Kilos, 9:44:20, in Hub
20, 3595 Main St, Salt Lake City, UT, 84115, 10:30 AM, 37 Kilos, 9:42:40, in Hub
21, 3595 Main St, Salt Lake City, UT, 84115, EOD, 3 Kilos, 8:06:40, in Hub
22, 6351 South 900 East, Murray, UT, 84121, EOD, 2 Kilos, 9:43:40, in Hub
23, 5100 South 2700 West, Salt Lake City, UT, 84118, EOD, 5 Kilos, 10:25:40, in Hub
24, 5025 State St, Murray, UT, 84107, EOD, 7 Kilos, 9:33:20, in Hub
25, 5383 South 900 East #104, Salt Lake City, UT, 84117, 10:30 AM, 7 Kilos, 9:29:20, in Hub
26, 5383 South 900 East #104, Salt Lake City, UT, 84117, EOD, 25 Kilos, 9:29:20, in Hub
27, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 5 Kilos, 10:18:40, in Hub
28, 2835 Main St, Salt Lake City, UT, 84115, EOD, 7 Kilos, 9:49:20, in Hub
29, 1330 2100 S, Salt Lake City, UT, 84106, 10:30 AM, 2 Kilos, 8:25:00, in Hub
30, 300 State St, Salt Lake City, UT, 84103, 10:30 AM, 1 Kilos, 8:42:40, in Hub
```



```
29, 1330 2100 S, Salt Lake City, UT, 84106, 10:30 AM, 2 Kilos, 8:25:00, in Hub
30, 300 State St, Salt Lake City, UT, 84103, 10:30 AM, 1 Kilos, 8:42:40, in Hub
31, 3365 S 900 W, Salt Lake City, UT, 84119, 10:30 AM, 1 Kilos, 9:16:00, in Hub
32, 3365 S 900 W, Salt Lake City, UT, 84119, EOD, 1 Kilos, 9:59:00, in Hub
33, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 1 Kilos, 10:29:20, in Hub
34, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 2 Kilos, 9:59:20, in Hub
35, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 88 Kilos, 11:09:20, in Hub
36, 2300 Parkway Blvd, West Valley City, UT, 84119, EOD, 88 Kilos, 10:09:20, in Hub
37, 410 S State St, Salt Lake City, UT, 84111, 10:30 AM, 2 Kilos, 8:39:20, in Hub
38, 410 S State St, Salt Lake City, UT, 84111, EOD, 9 Kilos, 10:34:40, in Hub
39, 2010 W 500 S, Salt Lake City, UT, 84104, EOD, 9 Kilos, 8:56:40, in Hub
40, 380 W 2880 S, Salt Lake City, UT, 84115, 10:30 AM, 45 Kilos, 8:12:00, in Hub

Process finished with exit code 0
```

Section I:

1)

Some strengths of the nearest neighbor algorithm are its intuitiveness and simplicity. The algorithm is easy to understand and implement so it provides ease of access for others to take over. To find the next data point, the program simply reads through the whole data set to find the next closest neighbor to travel to. Although not quite the best solution to find the shortest route for all trucks, it brings the overall distance well below the required minimum of 140 miles.

Another strength of this algorithm is its constant evolution/adaptability. The algorithm is able to work with any sized amount of data without taking much of a loss on speed or accuracy. It responds quickly to input change during real time use. So, for example if more addresses were added while the program was running it would be able to account for those as well to make sure it is still fining the nearest neighbor.

2)

When looking at the screenshots provided, we can see that all packages have been delivered on time, with special instructions fulfilled, and within the distance constraint giving us a total of 108.1 miles for the delivery. Special instructions include:

- Package 9 will update at 10:20 am and be delivered to the correct address.
- All packages start at the hub before departing from the hub
- Package 13, 14, 15, 16, 19, and 20 have all been delivered together.

3)

Two other algorithms that could have met the given requirements are Dijkstra's algorithm or a simple greedy algorithm. Both of which have the capability to bring the mileage below 140 and even below the current mileage of this program which was 108.1 miles.

A) Dijkstra's algorithm differs from the nearest neighbor algorithm as it has the distances to each location precalculated at the beginning whereas the nearest neighbor calculates distances as they come. Dijkstra's uses a graph data structure to build trees from a start node to other nodes in the graph and then records the shortest distance shortest path taken in the tree to get to the next node. The graph and start node are initialized first to begin the search where then everything else is calculated after.

B) The process of the greedy algorithm in simple terms is at each node in a graph, go to the nearest unvisited node in the graph. While this approach might work well in some cases, it can also lead to suboptimal solutions. For example, take the list of addresses that includes a city that is relatively close to the starting city, but it is not the next city on the list. In this case, the greedy algorithm would not visit that city until much later in the process. This will make the route longer than it could potentially be.

Section J:

If this project were to be completed again, I would change the process of assigning packages to trucks. For this program, the packages were manually loaded into the trucks within the code. In the long term this could be a problem if there are days with a lot more packages than in this given situation. This would be the biggest process to automate and have directly handled by the program to save time for the users.

Section K:

1)

The chaining hash table used in the program met all of the given requirements to fulfill. It delivered all of the packages by their deadlines at a total of 108.1 miles. All of the package information was accurately displayed with a search function and the total report of the delivery was quick and precise.

A) Managing key-value data in this kind of hash table is fairly efficient in this case as it uses a form of linear probing to access the data. The time complexity of this implementation will always be O(n) with the worst-case time only varying by the overall size of the data set. If there are more packages in the system it will take longer to parse and search for it.

B) Space optimization is a key characteristic in a chaining hash table. When a new element is added to the table, it is placed in the linked list at the index in the array corresponding to its hash value. This allows multiple elements to be stored at the same index in the array, reducing the amount of unused space in the table. It is important to make sure to provide enough space at initialization to maintain speed as it will slow down to allocate more space if there isn't enough.

C) Changing the number of trucks in this application will have little effect on package search time. All truck information is stored as an object where in this program package data isn't linked to any specific truck data. So, searching for a package will maintain its O(n) complexity and its space allocation will remain the same.

2)

Two other data structures that could have been implemented in place of the chaining hash table are stacks and binary trees.

A) A binary tree can be used to store the package addresses that the trucks would need to go to. Binary trees are data structures in which each node has at most two children. They are typically used to store data that is naturally hierarchical, such as the structure of a file system or the hierarchy of a company. Hash tables on the other hand do not store data in that way. Binary trees are well-suited for searching, as they allow for fast search times by allowing the search to be narrowed down as it progresses through the tree. However, they are not as efficient as hash tables for inserting and deleting data, as these operations may require the tree to be reorganized to maintain its balance.

B) Stacks can be used to automate the loading and unloading of trucks as the route progresses. A stack is a linear data structure that follows the Last in First Out principle, meaning that the most recently added element is the first one to be removed. Stacks are commonly used to store data temporarily which in this case would be packages. They also support two primary operations which are push, which adds an element to the top of the stack, and pop, which removes the element from the top of the stack. They do not provide efficient search or insertion/deletion performance like hash tables do for elements not at the top of the stack. A hash table can search directly for specific data while a stack has to pop items until the data is found.

Section L:

No sources have been quoted, paraphrased, or summarized in this assignment.