WGUPS Package Routing Software:
A Parallel Routing Approach

Steven Barton
Department of Information Technology – Computer Science
Data Structures and Algorithms - C950
Professor Kristopher Goodell

Abstract

Western Governors United Parcel Service (WGUPS) required software to manage the loading and routing of their three trucks in the Salt Lake City area. The trucks could hold 16 packages per load, maintained a constant speed of 18 miles per hour, and were not affected by collisions or stop times. At the time of programming two drivers were available. The map of delivery locations was represented with a graph data structure, the packages were stored in a hash-table data structure, and a parallel nearest neighbor algorithm was used to develop routes for the trucks which would result in a total mileage between all routes run of less than 170 miles in total.

Alternatives to the applied algorithm and package storage data structure are discussed. The algorithm alternatives discussed are Dijkstra's algorithm and the depth first search algorithm. The data structure alternatives discussed are the binary search tree and set data structures. Space time complexity is also analyzed for this program and compared with alternative algorithms.

A. **Self-Adjusting Algorithm**: I chose to use an alternative implementation of the nearest neighbor algorithm. This algorithm is a greedy algorithm which takes the nearest neighbor to each node as the path is created to develop a good-enough solution in a short amount of time. (Koether, 2016) This algorithm easily finds the shortest adjacent edge to an unvisited node using the adjacency matrix implementation I used for the graph. The program takes the packages to be loaded next and uses those to feed only the corresponding locations into the algorithm. My implementation of the algorithm starts by creating two path lists with the hub as the initial location. The algorithm takes the nearest location as the next location in the first list, then the second closest (closest unvisited) as the next location in the second list. From there on the nearest unvisited neighbor to the top item in each list is added until there are no more unvisited locations from the list. The second list is then added to the end of the first in reverse order to create an out and back route instead of simply navigating to the furthest point and driving all the way back. This list is returned as the route for the truck.

B. **Program Overview**:

1. Algorithm Logic:

ParallelRoute
The nearest neighbor algorithm which returns a short route for package delivery.

Input: A list of packages to be loaded in a truck.
Output: An ordered list of locations for the truck route.

Set all locations not corresponding to the package list as visited

Queue1 append hub location
Queue2 append hub location

While queue1 and queue2 are not empty:

Location1 = queue1 pop first value
Path1 append Location1
Location1 set visited

Location2 = queue2 pop first value
Path2 append Location2
Location2 set visited

Neighbor1 = Location1 closest unvisited node
Queue1.append Neighbor1

Neightbor2 = Location2 closest unvisited node

```
        Queue2.append Neighbor2

    Path = Path1

    For i in range Path2 length:
        Path append Path2[Path2 length – i]

    Return Path
```

2. Programming Environment:

   Operating System: macOS Big Sur Version 11.5.2
   Hardware: MacBook Pro, 2.3 GHz 8-Core Intel Core i9, 16 GB 2667 MHz DDR4
   IDE: PyCharm 2021.3.2 (Community Edition)
   Language: Python

3. Space-Time Complexity:

   Hash Table: $O(1)$ best case insertion, $O(N)$ worst case insertion, $O(1)$ best case access, $O(N)$ worst case access. (Lysecky, 2018)

   Graph: $O(N^2)$ insertion, $O(N)$ access. (Lysecky, 2018)

   Nearest Neighbor Algorithm: $O(N^2)$ (Koether, 2016)

   Queuing of Packages (address lookup and package sorting): $O(N^2)$

4. Capability to Scale: This program can easily adapt to a growing number of packages due to the self-adjusting nature of the hash table. The hash table uses a python list as the underlying storage for package objects. The hash table maintains a buffer of empty indices to facilitate quadratic probing which reduces clustering and quickly finds an empty index. When the limit of the ratio of stored package objects to list size is reached the hash table automatically increases the size of the underlying list and then re-hashes each item into the larger list. As long as physical memory is not filled the hash table will never fill up and be unable to accept package objects.

5. Efficiency and Ease of Maintenance: This program is designed with an object-oriented structure. Each portion of the program is separated into classes which define objects. The functions used are contained in the object on which they operate. If updates are necessary for any portion of the code these can easily be made within a class and function. If the class names, function names, and input/output parameters remain the same code can be updated as necessary to

streamline operation without having cascading errors in the code. This also allows debugging to be completed easily as errors can be traced back to their source function. Additionally, every class, function, and block of logic is labelled with comments which describe the purpose and logic of the section of code.

6. Strengths and Weaknesses of Data Structures:

Hash Table Strengths: The hash table has very fast insertion and access for packages. Hashing allows for real-time to near real-time insertion and access using the unique package id number as a key which is hashed into an index. The hash table is also self-adjusting so it will automatically scale itself to growing numbers of packages being inserted. The program will continue to function seamlessly as packages are inserted. (Lysecki, 2018)

Hash Table Weaknesses: The hash table requires extra memory to accommodate for the empty indices in the underlying list. The number of empty indices in the list will increase proportionally to the number of packages objects stored in the hash table. Another weakness is the time required every time the hash table requires a re-hash in order to grow. The re-hash requires $O(N)$ time in order to rehash all stored items. (Lysecky, 2018)

Graph Strengths: The graph data structure implemented in this program enables very fast access to edges due to its adjacency matrix structure. Accessing an edge is $O(1)$ requiring only the input of the index of the location on either end of the edge. This implementation also only requires edge weights be stored in a 2D list and does not require linked lists of objects be stored in the data structure. This reduces the size of each stored node and simplifies the code required to use the structure. (Lysecki, 2018)

Graph Weaknesses: The adjacency matrix implementation of the graph requires $O(N^2)$ time to build and $O(N)$ time to append. This is due to the fact that the graph is universally connected between nodes. Each node added requires an edge be added between it and every other node. The universally connected graph also requires N^2 edges be stored for N nodes stored in the graph. There is no way to reduce this with either an adjacency matrix or an adjacency list implementation. (Lysecki, 2018)

C. **Program Requirements**:

1. Main.py comment: Complete

2. Comments within code: Complete

D. **Package Storage Data Structure**: Hash Table

1. Accounting for Data Point Relationships: The hash table implemented here differentiates between packages using the unique package ID as a primary key. Packages may have identical address data, but they are assigned a unique ID when being stored. This is used to hash them into a list index in the hash table. Packages are easily inserted and searched. Each package is stored as an object which contains all of its associated data and allows data to be stored in discreet package objects which are easily searched.

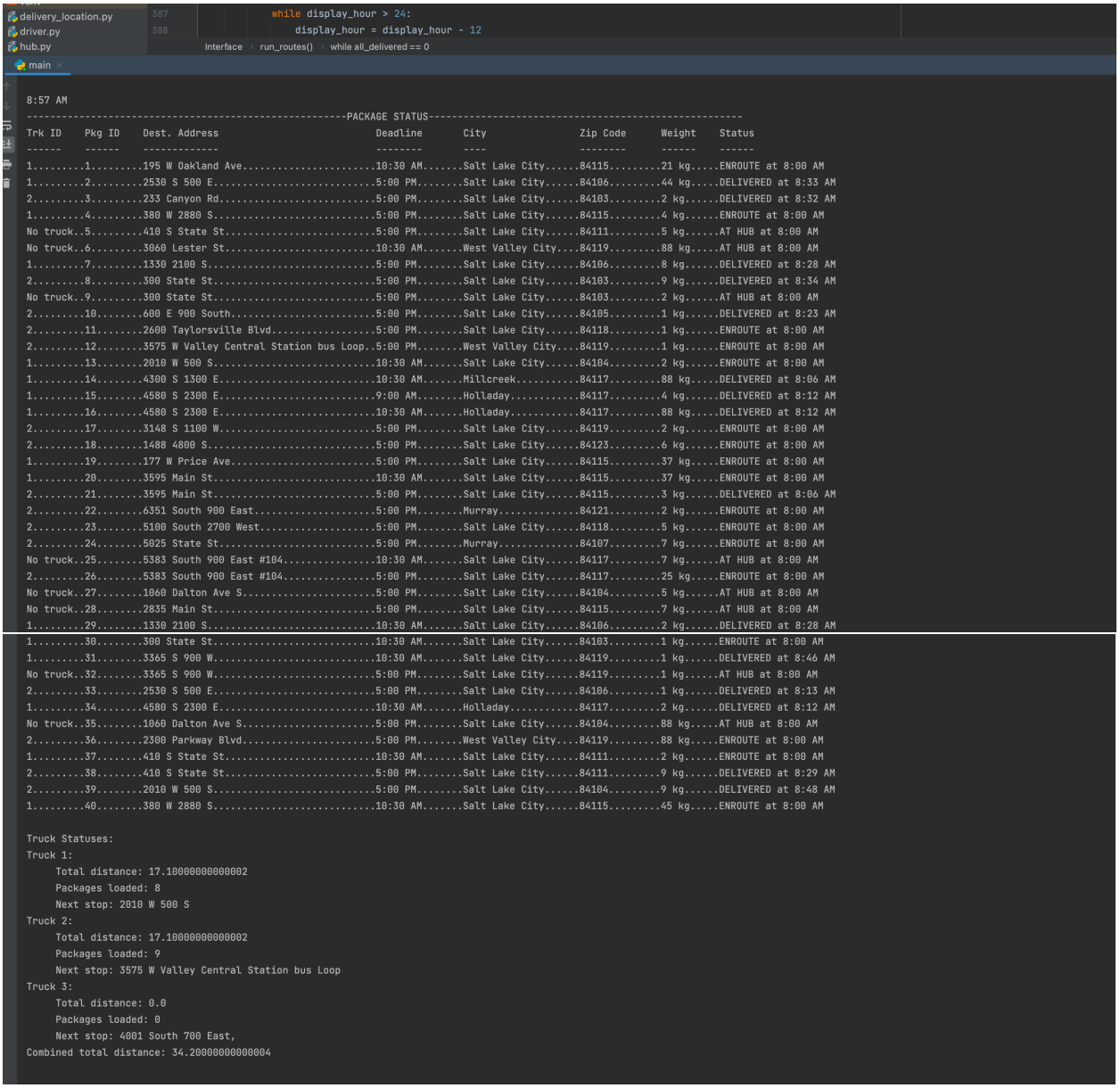
E. **Hash Table Insertion**: Hash table insertion is completed by inserting a package object. The package object has the constructor method call below:

wgups_package(id, address, deadline, city, zip_code, weight, status)

F. **Lookup Function**: The lookup function is written into the interface of the program. The lookup function accepts each data element contained in a package object. When search terms are input for any of the fields a list is created and printed of all packages that match the search terms.

G. **Interface Screenshots**: See subsequent pages.

1. Screenshot between 8:35 a.m. and 9:25 a.m.:

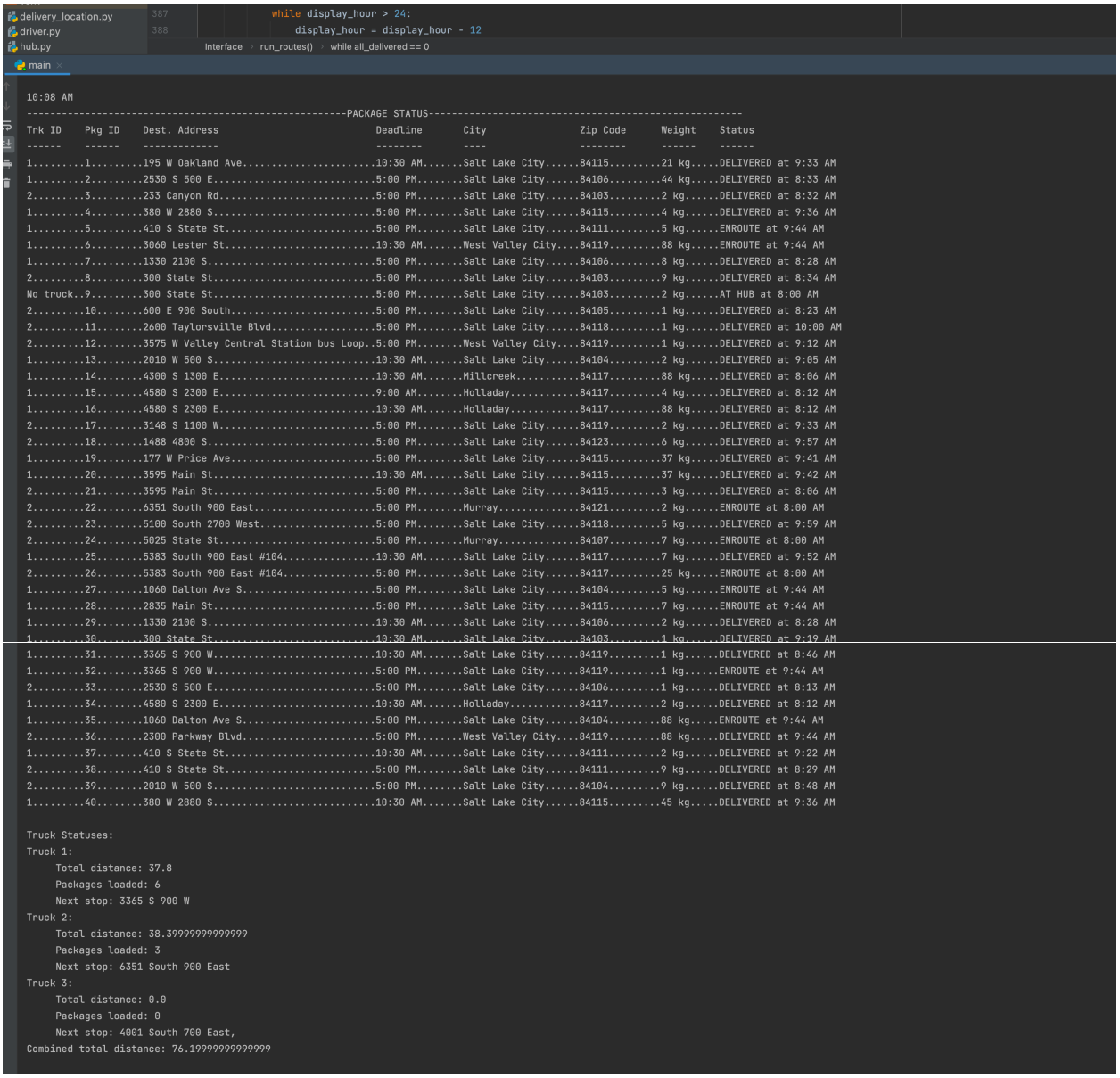


```
delivery_location.py   387        while display_hour > 24:
driver.py              388            display_hour = display_hour - 12
hub.py                 Interface › run_routes() › while all_delivered == 0
main

8:57 AM
-----------------------------------------------------------PACKAGE STATUS-----------------------------------------------------------
Trk ID    Pkg ID    Dest. Address                          Deadline       City               Zip Code      Weight    Status
------    ------    -------------                          --------       ----               --------      ------    ------
1.........1.........195 W Oakland Ave.....................10:30 AM.......Salt Lake City......84115.........21 kg.....ENROUTE at 8:00 AM
1.........2.........2530 S 500 E..........................5:00 PM........Salt Lake City......84106.........44 kg.....DELIVERED at 8:33 AM
2.........3.........233 Canyon Rd..........................5:00 PM........Salt Lake City......84103.........2 kg......DELIVERED at 8:32 AM
1.........4.........380 W 2880 S...........................5:00 PM........Salt Lake City......84115.........4 kg......ENROUTE at 8:00 AM
No truck..5.........410 S State St.........................5:00 PM........Salt Lake City......84111.........5 kg......AT HUB at 8:00 AM
No truck..6.........3060 Lester St.........................10:30 AM.......West Valley City....84119.........88 kg.....AT HUB at 8:00 AM
1.........7.........1330 2100 S...........................5:00 PM........Salt Lake City......84106.........8 kg......DELIVERED at 8:28 AM
2.........8.........300 State St...........................5:00 PM........Salt Lake City......84103.........9 kg......DELIVERED at 8:34 AM
No truck..9.........300 State St...........................5:00 PM........Salt Lake City......84103.........2 kg......AT HUB at 8:00 AM
2.........10........600 E 900 South........................5:00 PM........Salt Lake City......84105.........1 kg......DELIVERED at 8:23 AM
2.........11........2600 Taylorsville Blvd..................5:00 PM........Salt Lake City......84118.........1 kg......ENROUTE at 8:00 AM
2.........12........3575 W Valley Central Station bus Loop..5:00 PM........West Valley City....84119.........1 kg......ENROUTE at 8:00 AM
1.........13........2010 W 500 S...........................10:30 AM.......Salt Lake City......84104.........2 kg......ENROUTE at 8:00 AM
1.........14........4300 S 1300 E..........................10:30 AM.......Millcreek...........84117.........88 kg.....DELIVERED at 8:06 AM
1.........15........4580 S 2300 E..........................9:00 AM........Holladay...........84117.........4 kg......DELIVERED at 8:12 AM
1.........16........4580 S 2300 E..........................10:30 AM.......Holladay...........84117.........88 kg.....DELIVERED at 8:12 AM
2.........17........3148 S 1100 W..........................5:00 PM........Salt Lake City......84119.........2 kg......ENROUTE at 8:00 AM
2.........18........1488 4800 S...........................5:00 PM........Salt Lake City......84123.........6 kg......ENROUTE at 8:00 AM
1.........19........177 W Price Ave.........................5:00 PM........Salt Lake City......84115.........37 kg.....ENROUTE at 8:00 AM
1.........20........3595 Main St..........................10:30 AM.......Salt Lake City......84115.........37 kg.....ENROUTE at 8:00 AM
2.........21........3595 Main St..........................5:00 PM........Salt Lake City......84115.........3 kg......DELIVERED at 8:06 AM
2.........22........6351 South 900 East....................5:00 PM........Murray..............84121.........2 kg......ENROUTE at 8:00 AM
2.........23........5100 South 2700 West...................5:00 PM........Salt Lake City......84118.........5 kg......ENROUTE at 8:00 AM
2.........24........5025 State St..........................5:00 PM........Murray..............84107.........7 kg......ENROUTE at 8:00 AM
No truck..25........5383 South 900 East #104..............10:30 AM.......Salt Lake City......84117.........7 kg......AT HUB at 8:00 AM
2.........26........5383 South 900 East #104..............5:00 PM........Salt Lake City......84117.........25 kg.....ENROUTE at 8:00 AM
No truck..27........1060 Dalton Ave S.....................5:00 PM........Salt Lake City......84104.........5 kg......AT HUB at 8:00 AM
No truck..28........2835 Main St..........................5:00 PM........Salt Lake City......84115.........7 kg......AT HUB at 8:00 AM
1.........29........1330 2100 S...........................10:30 AM.......Salt Lake City......84106.........2 kg......DELIVERED at 8:28 AM
1.........30........300 State St...........................10:30 AM.......Salt Lake City......84103.........1 kg......ENROUTE at 8:00 AM
1.........31........3365 S 900 W...........................10:30 AM.......Salt Lake City......84119.........1 kg......DELIVERED at 8:46 AM
No truck..32........3365 S 900 W...........................5:00 PM........Salt Lake City......84119.........1 kg......AT HUB at 8:00 AM
2.........33........2530 S 500 E...........................5:00 PM........Salt Lake City......84106.........1 kg......DELIVERED at 8:13 AM
1.........34........4580 S 2300 E..........................10:30 AM.......Holladay...........84117.........2 kg......DELIVERED at 8:12 AM
No truck..35........1060 Dalton Ave S.....................5:00 PM........Salt Lake City......84104.........88 kg.....AT HUB at 8:00 AM
2.........36........2300 Parkway Blvd.....................5:00 PM........West Valley City....84119.........88 kg.....ENROUTE at 8:00 AM
1.........37........410 S State St.........................10:30 AM.......Salt Lake City......84111.........2 kg......ENROUTE at 8:00 AM
2.........38........410 S State St.........................5:00 PM........Salt Lake City......84111.........9 kg......DELIVERED at 8:29 AM
2.........39........2010 W 500 S...........................5:00 PM........Salt Lake City......84104.........9 kg......DELIVERED at 8:48 AM
1.........40........380 W 2880 S...........................10:30 AM.......Salt Lake City......84115.........45 kg.....ENROUTE at 8:00 AM

Truck Statuses:
Truck 1:
     Total distance: 17.10000000000002
     Packages loaded: 8
     Next stop: 2010 W 500 S
Truck 2:
     Total distance: 17.10000000000002
     Packages loaded: 9
     Next stop: 3575 W Valley Central Station bus Loop
Truck 3:
     Total distance: 0.0
     Packages loaded: 0
     Next stop: 4001 South 700 East,
Combined total distance: 34.20000000000004
```

2. Screenshot between 9:35 a.m. and 10:25 a.m.:

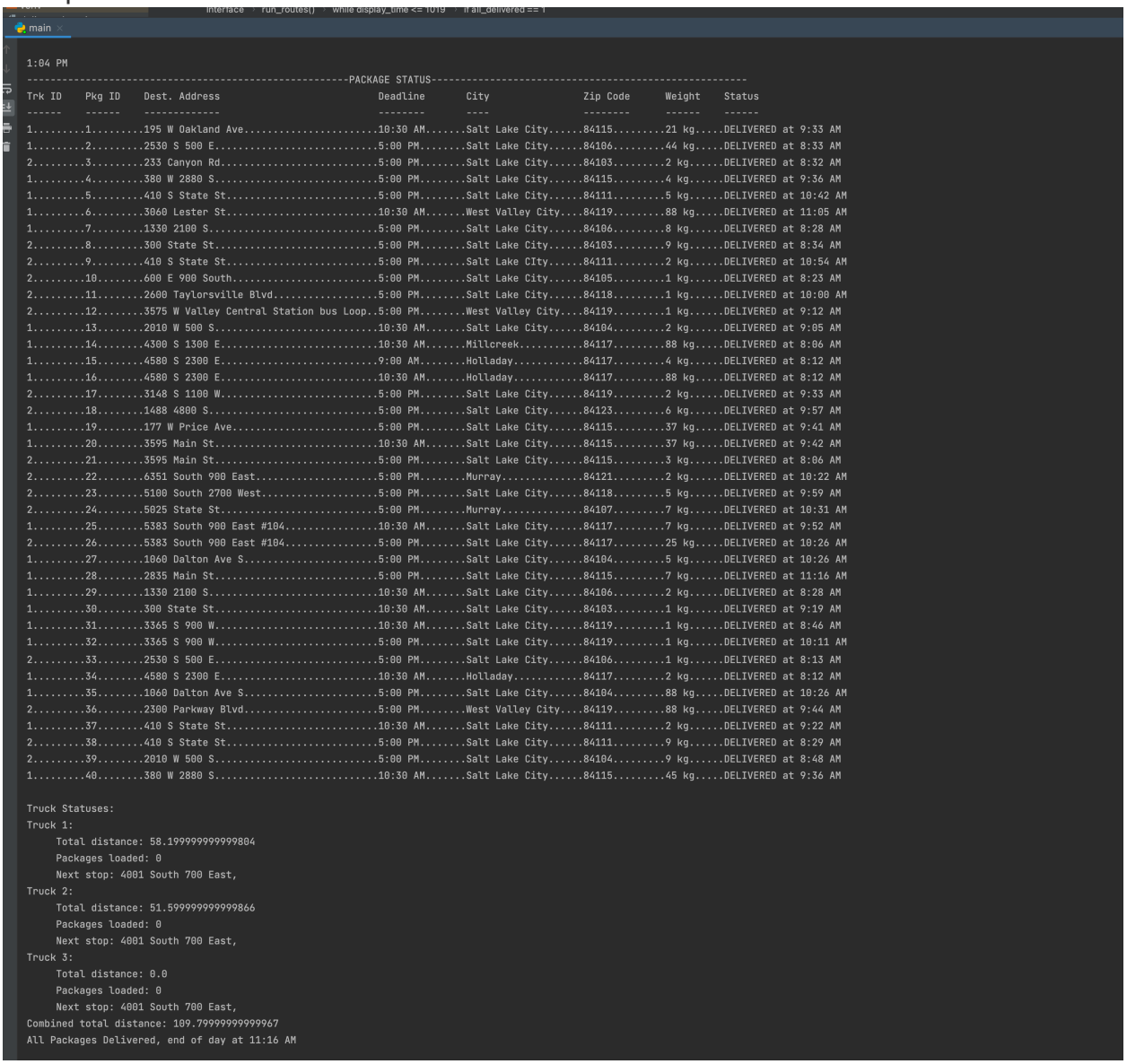


```
delivery_location.py   387        while display_hour > 24:
driver.py              388            display_hour = display_hour - 12
hub.py                 Interface › run_routes() › while all_delivered == 0
main

10:08 AM
-----------------------------------------------------------PACKAGE STATUS-----------------------------------------------------------
Trk ID    Pkg ID    Dest. Address                          Deadline       City               Zip Code      Weight    Status
------    ------    -------------                          --------       ----               --------      ------    ------
1.........1.........195 W Oakland Ave.....................10:30 AM.......Salt Lake City......84115.........21 kg.....DELIVERED at 9:33 AM
1.........2.........2530 S 500 E..........................5:00 PM........Salt Lake City......84106.........44 kg.....DELIVERED at 8:33 AM
2.........3.........233 Canyon Rd..........................5:00 PM........Salt Lake City......84103.........2 kg......DELIVERED at 8:32 AM
1.........4.........380 W 2880 S...........................5:00 PM........Salt Lake City......84115.........4 kg......DELIVERED at 9:36 AM
1.........5.........410 S State St.........................5:00 PM........Salt Lake City......84111.........5 kg......ENROUTE at 9:44 AM
1.........6.........3060 Lester St.........................10:30 AM.......West Valley City....84119.........88 kg.....ENROUTE at 9:44 AM
1.........7.........1330 2100 S...........................5:00 PM........Salt Lake City......84106.........8 kg......DELIVERED at 8:28 AM
2.........8.........300 State St..........................5:00 PM........Salt Lake City......84103.........9 kg......DELIVERED at 8:34 AM
No truck..9.........300 State St..........................5:00 PM........Salt Lake City......84103.........2 kg......AT HUB at 8:00 AM
2.........10........600 E 900 South........................5:00 PM........Salt Lake City......84105.........1 kg......DELIVERED at 8:23 AM
2.........11........2600 Taylorsville Blvd..................5:00 PM........Salt Lake City......84118.........1 kg......DELIVERED at 10:00 AM
2.........12........3575 W Valley Central Station bus Loop..5:00 PM........West Valley City....84119.........1 kg......DELIVERED at 9:12 AM
1.........13........2010 W 500 S...........................10:30 AM.......Salt Lake City......84104.........2 kg......DELIVERED at 9:05 AM
1.........14........4300 S 1300 E..........................10:30 AM.......Millcreek...........84117.........88 kg.....DELIVERED at 8:06 AM
1.........15........4580 S 2300 E..........................9:00 AM........Holladay...........84117.........4 kg......DELIVERED at 8:12 AM
1.........16........4580 S 2300 E..........................10:30 AM.......Holladay...........84117.........88 kg.....DELIVERED at 8:12 AM
2.........17........3148 S 1100 W..........................5:00 PM........Salt Lake City......84119.........2 kg......DELIVERED at 9:33 AM
2.........18........1488 4800 S...........................5:00 PM........Salt Lake City......84123.........6 kg......DELIVERED at 9:57 AM
1.........19........177 W Price Ave.........................5:00 PM........Salt Lake City......84115.........37 kg.....DELIVERED at 9:41 AM
1.........20........3595 Main St...........................10:30 AM.......Salt Lake City......84115.........37 kg.....DELIVERED at 9:42 AM
2.........21........3595 Main St...........................5:00 PM........Salt Lake City......84115.........3 kg......DELIVERED at 8:06 AM
2.........22........6351 South 900 East.....................5:00 PM........Murray..............84121.........2 kg......ENROUTE at 8:00 AM
2.........23........5100 South 2700 West....................5:00 PM........Salt Lake City......84118.........5 kg......DELIVERED at 9:59 AM
2.........24........5025 State St...........................5:00 PM........Murray..............84107.........7 kg......ENROUTE at 8:00 AM
1.........25........5383 South 900 East #104...............10:30 AM.......Salt Lake City......84117.........7 kg......DELIVERED at 9:52 AM
2.........26........5383 South 900 East #104...............5:00 PM........Salt Lake City......84117.........25 kg.....ENROUTE at 8:00 AM
1.........27........1060 Dalton Ave S.....................5:00 PM........Salt Lake City......84104.........5 kg......ENROUTE at 9:44 AM
1.........28........2835 Main St...........................5:00 PM........Salt Lake City......84115.........7 kg......ENROUTE at 9:44 AM
1.........29........1330 2100 S...........................10:30 AM.......Salt Lake City......84106.........2 kg......DELIVERED at 8:28 AM
1.........30........300 State St...........................10:30 AM.......Salt Lake City......84103.........1 kg......DELIVERED at 9:19 AM
1.........31........3365 S 900 W...........................10:30 AM.......Salt Lake City......84119.........1 kg......DELIVERED at 8:46 AM
1.........32........3365 S 900 W...........................5:00 PM........Salt Lake City......84119.........1 kg......ENROUTE at 9:44 AM
2.........33........2530 S 500 E...........................5:00 PM........Salt Lake City......84106.........1 kg......DELIVERED at 8:13 AM
1.........34........4580 S 2300 E..........................10:30 AM.......Holladay...........84117.........2 kg......DELIVERED at 8:12 AM
1.........35........1060 Dalton Ave S.....................5:00 PM........Salt Lake City......84104.........88 kg.....ENROUTE at 9:44 AM
2.........36........2300 Parkway Blvd.....................5:00 PM........West Valley City....84119.........88 kg.....DELIVERED at 9:44 AM
1.........37........410 S State St.........................10:30 AM.......Salt Lake City......84111.........2 kg......DELIVERED at 9:22 AM
2.........38........410 S State St.........................5:00 PM........Salt Lake City......84111.........9 kg......DELIVERED at 8:29 AM
2.........39........2010 W 500 S...........................5:00 PM........Salt Lake City......84104.........9 kg......DELIVERED at 8:48 AM
1.........40........380 W 2880 S...........................10:30 AM.......Salt Lake City......84115.........45 kg.....DELIVERED at 9:36 AM

Truck Statuses:
Truck 1:
     Total distance: 37.8
     Packages loaded: 6
     Next stop: 3365 S 900 W
Truck 2:
     Total distance: 38.39999999999999
     Packages loaded: 3
     Next stop: 6351 South 900 East
Truck 3:
     Total distance: 0.0
     Packages loaded: 0
     Next stop: 4001 South 700 East,
Combined total distance: 76.19999999999999
```

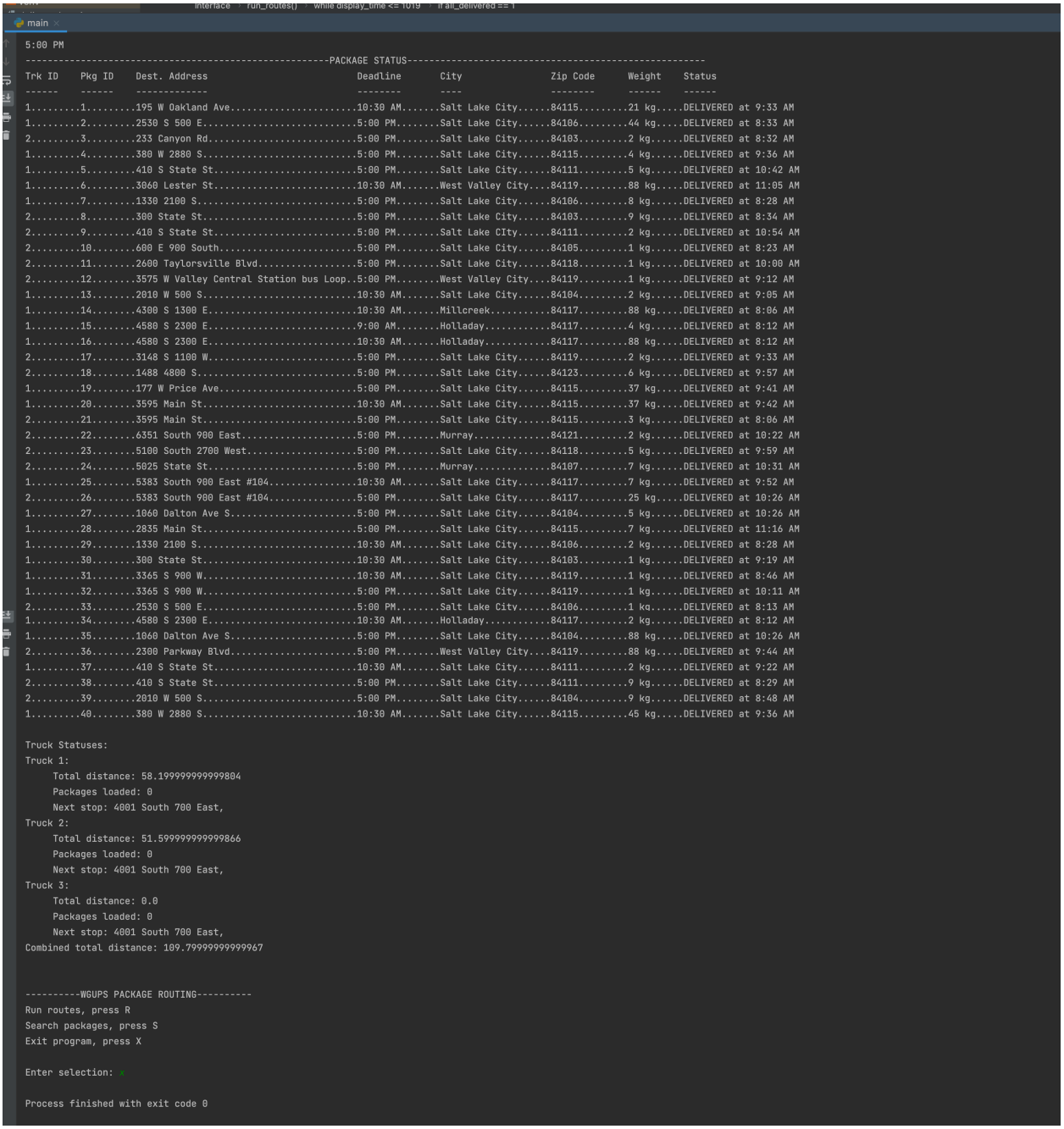
3. Screenshot to show the status of *all* packages at a time between 12:03 p.m. and 1:12 p.m.:



```
1:04 PM
------------------------------------------------------------PACKAGE STATUS------------------------------------------------------------
Trk ID    Pkg ID    Dest. Address                          Deadline       City               Zip Code      Weight    Status
------    ------    -------------                          --------       ----               --------      ------    ------
1.........1.........195 W Oakland Ave.....................10:30 AM.......Salt Lake City......84115.........21 kg.....DELIVERED at 9:33 AM
1.........2.........2530 S 500 E..........................5:00 PM........Salt Lake City......84106.........44 kg.....DELIVERED at 8:33 AM
2.........3.........233 Canyon Rd..........................5:00 PM........Salt Lake City......84103.........2 kg......DELIVERED at 8:32 AM
1.........4.........380 W 2880 S...........................5:00 PM........Salt Lake City......84115.........4 kg......DELIVERED at 9:36 AM
1.........5.........410 S State St.........................5:00 PM........Salt Lake City......84111.........5 kg......DELIVERED at 10:42 AM
1.........6.........3060 Lester St.........................10:30 AM.......West Valley City....84119.........88 kg.....DELIVERED at 11:05 AM
1.........7.........1330 2100 S...........................5:00 PM........Salt Lake City......84106.........8 kg......DELIVERED at 8:28 AM
2.........8.........300 State St..........................5:00 PM........Salt Lake City......84103.........9 kg......DELIVERED at 8:34 AM
2.........9.........410 S State St.........................5:00 PM........Salt Lake CIty......84111.........2 kg......DELIVERED at 10:54 AM
2.........10........600 E 900 South........................5:00 PM........Salt Lake City......84105.........1 kg......DELIVERED at 8:23 AM
2.........11........2600 Taylorsville Blvd..................5:00 PM........Salt Lake City......84118.........1 kg......DELIVERED at 10:00 AM
2.........12........3575 W Valley Central Station bus Loop..5:00 PM........West Valley City....84119.........1 kg......DELIVERED at 9:12 AM
1.........13........2010 W 500 S...........................10:30 AM.......Salt Lake City......84104.........2 kg......DELIVERED at 9:05 AM
1.........14........4300 S 1300 E..........................10:30 AM.......Millcreek...........84117.........88 kg.....DELIVERED at 8:06 AM
1.........15........4580 S 2300 E..........................9:00 AM........Holladay............84117.........4 kg......DELIVERED at 8:12 AM
1.........16........4580 S 2300 E..........................10:30 AM.......Holladay............84117.........88 kg.....DELIVERED at 8:12 AM
2.........17........3148 S 1100 W..........................5:00 PM........Salt Lake City......84119.........2 kg......DELIVERED at 9:33 AM
2.........18........1488 4800 S...........................5:00 PM........Salt Lake City......84123.........6 kg......DELIVERED at 9:57 AM
1.........19........177 W Price Ave.........................5:00 PM........Salt Lake City......84115.........37 kg.....DELIVERED at 9:41 AM
1.........20........3595 Main St...........................10:30 AM.......Salt Lake City......84115.........37 kg.....DELIVERED at 9:42 AM
2.........21........3595 Main St...........................5:00 PM........Salt Lake City......84115.........3 kg......DELIVERED at 8:06 AM
2.........22........6351 South 900 East....................5:00 PM........Murray..............84121.........2 kg......DELIVERED at 10:22 AM
2.........23........5100 South 2700 West....................5:00 PM........Salt Lake City......84118.........5 kg......DELIVERED at 9:59 AM
2.........24........5025 State St...........................5:00 PM........Murray..............84107.........7 kg......DELIVERED at 10:31 AM
1.........25........5383 South 900 East #104...............10:30 AM.......Salt Lake City......84117.........7 kg......DELIVERED at 9:52 AM
2.........26........5383 South 900 East #104...............5:00 PM........Salt Lake City......84117.........25 kg.....DELIVERED at 10:26 AM
1.........27........1060 Dalton Ave S.....................5:00 PM........Salt Lake City......84104.........5 kg......DELIVERED at 10:26 AM
1.........28........2835 Main St..........................5:00 PM........Salt Lake City......84115.........7 kg......DELIVERED at 11:16 AM
1.........29........1330 2100 S...........................10:30 AM.......Salt Lake City......84106.........2 kg......DELIVERED at 8:28 AM
1.........30........300 State St..........................10:30 AM.......Salt Lake City......84103.........1 kg......DELIVERED at 9:19 AM
1.........31........3365 S 900 W..........................10:30 AM.......Salt Lake City......84119.........1 kg......DELIVERED at 8:46 AM
1.........32........3365 S 900 W..........................5:00 PM........Salt Lake City......84119.........1 kg......DELIVERED at 10:11 AM
2.........33........2530 S 500 E..........................5:00 PM........Salt Lake City......84106.........1 kg......DELIVERED at 8:13 AM
1.........34........4580 S 2300 E..........................10:30 AM.......Holladay............84117.........2 kg......DELIVERED at 8:12 AM
1.........35........1060 Dalton Ave S.....................5:00 PM........Salt Lake City......84104.........88 kg.....DELIVERED at 10:26 AM
2.........36........2300 Parkway Blvd.....................5:00 PM........West Valley City....84119.........88 kg.....DELIVERED at 9:44 AM
1.........37........410 S State St.........................10:30 AM.......Salt Lake City......84111.........2 kg......DELIVERED at 9:22 AM
2.........38........410 S State St.........................5:00 PM........Salt Lake City......84111.........9 kg......DELIVERED at 8:29 AM
2.........39........2010 W 500 S...........................5:00 PM........Salt Lake City......84104.........9 kg......DELIVERED at 8:48 AM
1.........40........380 W 2880 S...........................10:30 AM.......Salt Lake City......84115.........45 kg.....DELIVERED at 9:36 AM

Truck Statuses:
Truck 1:
     Total distance: 58.199999999999804
     Packages loaded: 0
     Next stop: 4001 South 700 East,
Truck 2:
     Total distance: 51.599999999999866
     Packages loaded: 0
     Next stop: 4001 South 700 East,
Truck 3:
     Total distance: 0.0
     Packages loaded: 0
     Next stop: 4001 South 700 East,
Combined total distance: 109.79999999999967
All Packages Delivered, end of day at 11:16 AM
```

H. **Screenshot showing successful completion**:



```
main

5:00 PM
-----------------------------------------------------------PACKAGE STATUS-----------------------------------------------------------
Trk ID    Pkg ID    Dest. Address                          Deadline       City               Zip Code      Weight    Status
------    ------    -------------                          --------       ----               --------      ------    ------
1.........1.........195 W Oakland Ave.....................10:30 AM.......Salt Lake City......84115.........21 kg.....DELIVERED at 9:33 AM
1.........2.........2530 S 500 E..........................5:00 PM........Salt Lake City......84106.........44 kg.....DELIVERED at 8:33 AM
2.........3.........233 Canyon Rd..........................5:00 PM........Salt Lake City......84103.........2 kg......DELIVERED at 8:32 AM
1.........4.........380 W 2880 S...........................5:00 PM........Salt Lake City......84115.........4 kg......DELIVERED at 9:36 AM
1.........5.........410 S State St.........................5:00 PM........Salt Lake City......84111.........5 kg......DELIVERED at 10:42 AM
1.........6.........3060 Lester St.........................10:30 AM.......West Valley City....84119.........88 kg.....DELIVERED at 11:05 AM
1.........7.........1330 2100 S...........................5:00 PM........Salt Lake City......84106.........8 kg......DELIVERED at 8:28 AM
2.........8.........300 State St...........................5:00 PM........Salt Lake City......84103.........9 kg......DELIVERED at 8:34 AM
2.........9.........410 S State St.........................5:00 PM........Salt Lake CIty......84111.........2 kg......DELIVERED at 10:54 AM
2.........10........600 E 900 South........................5:00 PM........Salt Lake City......84105.........1 kg......DELIVERED at 8:23 AM
2.........11........2600 Taylorsville Blvd..................5:00 PM........Salt Lake City......84118.........1 kg......DELIVERED at 10:00 AM
2.........12........3575 W Valley Central Station bus Loop..5:00 PM........West Valley City....84119.........1 kg......DELIVERED at 9:12 AM
1.........13........2010 W 500 S...........................10:30 AM.......Salt Lake City......84104.........2 kg......DELIVERED at 9:05 AM
1.........14........4300 S 1300 E..........................10:30 AM.......Millcreek...........84117.........88 kg.....DELIVERED at 8:06 AM
1.........15........4580 S 2300 E..........................9:00 AM........Holladay...........84117.........4 kg......DELIVERED at 8:12 AM
1.........16........4580 S 2300 E..........................10:30 AM.......Holladay...........84117.........88 kg.....DELIVERED at 8:12 AM
2.........17........3148 S 1100 W..........................5:00 PM........Salt Lake City......84119.........2 kg......DELIVERED at 9:33 AM
2.........18........1488 4800 S...........................5:00 PM........Salt Lake City......84123.........6 kg......DELIVERED at 9:57 AM
1.........19........177 W Price Ave.........................5:00 PM........Salt Lake City......84115.........37 kg.....DELIVERED at 9:41 AM
1.........20........3595 Main St...........................10:30 AM.......Salt Lake City......84115.........37 kg.....DELIVERED at 9:42 AM
2.........21........3595 Main St...........................5:00 PM........Salt Lake City......84115.........3 kg......DELIVERED at 8:06 AM
2.........22........6351 South 900 East....................5:00 PM........Murray.............84121.........2 kg......DELIVERED at 10:22 AM
2.........23........5100 South 2700 West...................5:00 PM........Salt Lake City......84118.........5 kg......DELIVERED at 9:59 AM
2.........24........5025 State St...........................5:00 PM........Murray.............84107.........7 kg......DELIVERED at 10:31 AM
1.........25........5383 South 900 East #104...............10:30 AM.......Salt Lake City......84117.........7 kg......DELIVERED at 9:52 AM
2.........26........5383 South 900 East #104...............5:00 PM........Salt Lake City......84117.........25 kg.....DELIVERED at 10:26 AM
1.........27........1060 Dalton Ave S.....................5:00 PM........Salt Lake City......84104.........5 kg......DELIVERED at 10:26 AM
1.........28........2835 Main St...........................5:00 PM........Salt Lake City......84115.........7 kg......DELIVERED at 11:16 AM
1.........29........1330 2100 S...........................10:30 AM.......Salt Lake City......84106.........2 kg......DELIVERED at 8:28 AM
1.........30........300 State St...........................10:30 AM.......Salt Lake City......84103.........1 kg......DELIVERED at 9:19 AM
1.........31........3365 S 900 W..........................10:30 AM.......Salt Lake City......84119.........1 kg......DELIVERED at 8:46 AM
1.........32........3365 S 900 W..........................5:00 PM........Salt Lake City......84119.........1 kg......DELIVERED at 10:11 AM
2.........33........2530 S 500 E..........................5:00 PM........Salt Lake City......84106.........1 kg......DELIVERED at 8:13 AM
1.........34........4580 S 2300 E..........................10:30 AM.......Holladay...........84117.........2 kg......DELIVERED at 8:12 AM
1.........35........1060 Dalton Ave S.....................5:00 PM........Salt Lake City......84104.........88 kg.....DELIVERED at 10:26 AM
2.........36........2300 Parkway Blvd.....................5:00 PM........West Valley City....84119.........88 kg.....DELIVERED at 9:44 AM
1.........37........410 S State St.........................10:30 AM.......Salt Lake City......84111.........2 kg......DELIVERED at 9:22 AM
2.........38........410 S State St.........................5:00 PM........Salt Lake City......84111.........9 kg......DELIVERED at 8:29 AM
2.........39........2010 W 500 S...........................5:00 PM........Salt Lake City......84104.........9 kg......DELIVERED at 8:48 AM
1.........40........380 W 2880 S...........................10:30 AM.......Salt Lake City......84115.........45 kg.....DELIVERED at 9:36 AM

Truck Statuses:
Truck 1:
     Total distance: 58.199999999999804
     Packages loaded: 0
     Next stop: 4001 South 700 East,
Truck 2:
     Total distance: 51.599999999999866
     Packages loaded: 0
     Next stop: 4001 South 700 East,
Truck 3:
     Total distance: 0.0
     Packages loaded: 0
     Next stop: 4001 South 700 East,
Combined total distance: 109.79999999999967


----------WGUPS PACKAGE ROUTING----------
Run routes, press R
Search packages, press S
Exit program, press X

Enter selection: x

Process finished with exit code 0
```

I. **Core Algorithm Justification**:

1. Two Strengths of the Nearest Neighbor Algorithm: The nearest neighbor algorithm used provides a quick solution to the travelling salesman problem. For a list of N locations provided to the algorithm, a solution is returned in $O(N^2)$ time. This algorithm also returns a short path since it uses a greedy approach to make each leg of the path go along the shortest available edge. Since it is a greedy algorithm, the path is not necessarily the shortest path, but it is a short enough path. (Koether, 2016)

2. Algorithm meets requirements: The algorithm delivers all packages with a final distance of 109.8 miles combined between both trucks.

3. Two Alternatives: Dijkstra's Algorithm, Depth First Search

    a. How Each Algorithm is Different:

    Dijkstra's Algorithm: Dijkstra's algorithm takes the minimum path weight via all paths instead of just looking at the weight of the adjacent edges. This algorithm has the capability to calculate a shorter point to point path than the nearest neighbor algorithm since it takes each whole path into account. This algorithm was not used in this program because it requires a source and a destination node be provided which created problems including all nodes from the provided list in the path. (Lysecky, 2018)

    Depth First Search: The depth first search traverses all paths of a graph until a destination is reached instead of just looking at the adjacent edges. The depth first search can also find a shortest path by comparing edge weights similar to Dijkstra's algorithm, but it does not use previous node pointers. Depth first search can traverse all nodes easily but also requires a source node and a destination node. This created issues trying to do a round trip route. (Lysecki, 2018)

J. **What I would do differently**: If I wrote this program again, I would use an adjacency list to avoid having a matrix as well as a reference list of location objects on the side. Access to the matrix was simple and fast but I had to cross reference between matrix indices and the reference list of locations. If I used an adjacency list the first node in each linked list would contain the vertex with adjacent edges as subsequent list items. I would not need to cross reference between the matrix and the list of locations. Writing an adjacency list would require a linked list class which includes node objects with pointers to adjacent nodes. It would simplify use of algorithms like Dijkstra's algorithm because nodes allow for next and previous node pointers contained in a purpose build object.

K. **Data Structure Justification**:

1. Requirements Verification:

   a. Time Complexity with Increased Package Numbers: The lookup function in the interface can use any of the data fields for the package objects to look up packages. If the package ID is not provided the lookup function will iterate through each ID to match search terms. The basis of all lookup actions is the package ID which is used to retrieve packages from the hash table. If another search term is used, items are retrieved simply by iterating through package ID numbers. When searching using the package ID, which is the fastest and best method, searching is not significantly affected by increasing numbers of packages. Access via package ID uses quadratic probing to look for an ID that matches the supplied search term ID. No matter how many packages are inserted, searching remains best case $O(1)$ and worst case $O(N)$. (Lysecky, 2018)

   b. Space Usage with Increased Package Numbers: The space required for the hash table increases exponentially as packages are added due to the percentage-based buffer of empty indices which allow for quadratic probing in the event of collisions during insertion. As the buffer is reached, the hash table re-hashes its items into a larger list which has a percentage-based number of empty indices.

   c. Changes in Trucks and Locations Effects on Space and Time:

      Changes in the number of trucks will not affect the time or space of the hash-table. Packages are queued for loading in trucks using a list of their ID numbers. The packages are also logged as loaded into trucks using package ID search access. As trucks are loaded and packages are delivered, they are accessed with the previously mentioned time complexity to update their status.

      A change in the number of cities will affect the time required to load the packages since the address of each package is retrieved based on package ID and compared to locations so that they can be added to the route list. This is unavoidable and does not affect the actual space time complexity of the hash table itself.

2. Two Alternative Data Structures: Binary Search Tree, Set

   a. Binary Search Tree: A binary search tree does not hash a key to determine where to store an item in the data structure. The binary search tree would take the key and begin comparing it to nodes in the tree starting from the root node. Every insertion would move to the left or right child, left if the key is less than the node and right if more, until the node does not have a child where the key

value is going. In that case a leaf node child would be created for the key. Searches would work the same, returning the matching node, or if a leaf node is reached with no match the key is not present. Both insertions and searches would take longer on average with a time complexity of $O(\log N)$ for both. (Lysecky, 2018) Slightly less space would be used since no empty nodes are present as with a hash table for probing.

b. Set: A set uses a key for insertion but does not hash the key to find an index. The set is simply an unordered group of unique items. The set would also use a list as a base data structure on which to build, but items would simply be added to the end. Insertion would not be as fast since items must be unique, the complexity of insertion would be $O(N)$ for every insertion. Comparisons would need to be made with every element before inserting to ensure unique items are inserted. Searches would similarly be $O(N)$ on the worst case to compare values and find the desired key. Average case searching would depend on the algorithm used to traverse the list. (Lysecky, 2018)

Bibliography

Koether, R. T. (2016, November 16). *The traveling salesman problem nearest-neighbor algorithm - h-SC*. Robb Koether's Web Page. Retrieved June 28, 2022, from http://people.hsc.edu/faculty-staff/robbk/Math111/Lectures/Fall%202016/Lecture%2033%20-%20The%20Nearest-Neighbor%20Algorithm.pdf

Lysecky, R. (2018). 6.1 Graphs: Introduction. In F. Vahid (Ed.), *C950: Data Structures and Algorithms II* (June 2018). essay, Zyante Inc. Retrieved June 27, 2022, from https://learn.zybooks.com/zybook/WGUC950AY20182019/chapter/6/section/1.

Lysecky, R. (2018). 6.8 Graphs: Depth-first search. In F. Vahid (Ed.), *C950: Data Structures and Algorithms II* (June 2018). essay, Zyante Inc. Retrieved June 27, 2022, from https://learn.zybooks.com/zybook/WGUC950AY20182019/chapter/6/section/8.

Lysecky, R. (2018). 6.11 Algorithm: Dijkstra's shortest path. In F. Vahid (Ed.), *C950: Data Structures and Algorithms II* (June 2018). essay, Zyante Inc. Retrieved June 27, 2022, from https://learn.zybooks.com/zybook/WGUC950AY20182019/chapter/6/section/11.

Lysecky, R. (2018). 7.1 Hash Tables. In F. Vahid (Ed.), *C950: Data Structures and Algorithms II* (June 2018). essay, Zyante Inc. Retrieved June 27, 2022, from https://learn.zybooks.com/zybook/WGUC950AY20182019/chapter/7/section/1.