

Alex Irvine  
Student # 000955107  
2020/7/15  
WGU - C950 Project Submission

## Program Overview

The purpose of this program is to find the best route to deliver 40 packages for the fictional company of Western Governors University Parcel Service (WGUPS). These 40 packages are to be split up between three trucks and 2 drivers, each truck holding no more than 16 packages at one time. Many of the packages have special requirements or restrictions such as being limited to a certain truck, having a delayed start time, or containing the wrong address information. All packages must be delivered on time with all special requirements being met using an implemented algorithm to dynamically create the route.

The program is broken down into 4 main files:

**Package.py:** The package file holds the Package dataclass that is used to store information about each attribute associated with it. The `__str__` and `__eq__` methods are also overridden in order for better printouts in the CLI & to compare packages against their ID's.

**HashTable.py:** This class is responsible for holding all of the packages in the form of a direct hash table. Each bucket of the direct hash table represents a possible package ID and will store the package with the associated ID. Additionally a two-dimensional matrix of distances between each address and a counter of unique addresses are stored in this file. All package, address and distance data is retrieved from CSV files found in the data directory.

Methods for this file include basic CRUD (create, read, update, delete) commands, formatted output, and the instructions of how to separate each package into appropriate trucks.

**Truck.py:** The truck class is responsible for creating and executing the delivery route for the packages passed into it. Each truck has attributes for total millage driven, total time driven, current status, current position, and miles from the next position. A list of packages is dynamically sorted into an efficient delivery order using an implementation of Primm's algorithm (described in later sections).

**Main.py:** Main is the entry point for the user and contains the CLI used to interact with the program. Upon startup a global time variable is created and set to 8 AM, trucks are initialized and loaded with their packages, and the user is prompted to pick from a list of 10 options. The user has the ability to change the time of day, insert packages, search for packages, and receive a detailed printout of package and truck status details.

Each method in the program has detailed comments and docstrings in order to explain the logic and improve maintainability. A summary table of the space/time complexity of each method can be found near the end of this paper.

## Data Structure Overview

The main data structure used in the project is a direct hash table. A direct hash was chosen as it is the fastest available hash table. The downside to direct hashing is that it also creates the largest hash table, as each bucket can only store one package. However, since the company averages only 40 packages per day, this will still create a small memory footprint. The hash table by default is initialized with 100 buckets, but this can be

increased/decreased by changing the capacity when initializing the table.

When a package is added to the table from the CLI, a package ID is dynamically created based on available space in the hash table. When a free bucket is found the package ID will be set to it's index, and that package will be stored in that bucket. The address of the new package is also checked against current addresses, and the address ID is assigned if found. If no matching address is found, then a new address ID is created. This allows the hash table to self adjust to new packages created at run time.

Regardless of it's size the direct hash table will have a Big  $O(1)$  for package insert, deletion, and lookup by ID. If another package attribute is used for lookup instead of ID (such as city name), the direct hash table will have an efficiency of Big  $O(N)$  as it will need to perform a linear search through all buckets. The size of the direct hash table will be  $N$ , where  $N$  is the number of packages. Since most of the methods used on the direct hash table relies on package ID, the speed of runtime will not be affected since those methods run at Big  $O(1)$ . The only functionality of the program that will be affected is lookup by attributes other than ID (city, address, zip, weight, deadline, status), as that runs at Big  $O(N)$ . Using these functions will see a slowdown when more packages are added.

## Application of Programming Model

The application of this model is currently limited to being run on a local machine with Python 3.8 installed. There is also no communication protocol used since the program retrieves it's data from CSV files found in the program directory. Since data exchange is limited to the local machine, there is no target host environment outside of one running Python 3.8.

To read the data from CSV files, the CSV Reader module is used from Python's standard library. Package data is converted from the CSV file `CSV_DATA/packages.csv` and distances between addresses are converted from the CSV file `CSV_DATA/distances-filled.csv`. Each row of `packages.csv` contains the attributes for a package, and each row of `distances-filled.csv` contain the distances from one address to all other addresses. Each CSV file is iterated through one row at a time. A package class is created and inserted into the hash table for each row in `packages.csv`, and a row in the adjacency matrix of addresses is added for each row in `distances-filled.csv`. Currently the CSV file names are hard coded, but minor adjustments could be made to have the user select the csv files to pull from.

## Algorithm Overview

The algorithm to load the trucks and create the delivery route is as follows:

### Truck loading

1. The packages are first sorted based on the conditions of their requirements. Delayed packages, packages that must be delivered on truck 2, packages that must be delivered with other packages and urgent packages are separated into groups. The remaining packages are then placed into their own group.
2. Packages that must be placed on truck 2 are as well as packages that go to the same address. Urgent packages are also placed on truck 2.
3. Delayed packages are placed on truck 1 as well as packages going to the same address.
4. If there is remaining space on truck 1 or 2, packages with the closest address to an existing address on the truck are added until the truck is full.

5. All remaining packages are placed on truck 3.

## Route planning

1. Each truck starts at the hub (address 0).
2. Primm's path finding algorithm is used to create a minimal spanning tree:
  1. Two counters are created: num\_address & num\_edges.
  2. A list selected is created with length equal to num\_address.
  3. The first address index (hub) in selected is set to True.
  4. The program loops, checking the distance between each address to all other address. It stores the smallest distance between the addresses if one of the addresses is currently in selected.
  5. Once the smallest distance is found, an edge is created with data on the from address, to address and distance between. The edge is stored, and the to address is added to selected.
  6. Steps 4 & 5 repeat until num\_edges is equal to num\_address + 1.
3. A depth first search algorithm is then used to find a route through the minimal spanning tree created in the previous step:
  1. A placeholder for current address and two lists are created: visited and unvisited.
  2. The visited has the hub address added, the unvisited has all other addresses added.
  3. The program checks each edge of the minimum spanning tree. If the from address in the edge matches the current address and the to address is not in the visited list, the to address is added to visited and current address is updated to that address.
  4. If no addresses meet the criteria of the previous step, the current address is a leaf in the tree and the program backtracks, adding the address it's back tracking to into the visited list and updating the current address.
  5. Steps 3 & 4 repeat until there are no addresses left in the unvisited list. This will produce a path through the minimal spanning tree.
4. The path found in the previous step is then converted from a list into a dictionary, then back into a list. This removes all duplicate addresses from the list while preserving its order.
5. The hub address is then appended to the end of the path creating a Hamiltonian Cycle through the minimum spanning tree. This is the route to deliver packages.

## Advantages of Chosen Algorithm

The advantages of the chosen methods are mainly flexibility and scalability. By binding the path finding algorithms (Primm's and DFS) to the individual trucks we are to calculate the efficient paths independently for each truck. Since the path finding algorithms are the slowest part at Big  $O(N^2)$ , we can choose to run each truck's algorithms at different times from each other. If a bottle neck is later created when scaling up operations, the program can either stagger the path finding algorithms or potentially run them in parallel with adjustments.

## Alternate Data Structures

The program currently uses a direct hash table, but a chaining hash table could also be used to save memory space as it allows more packages to be placed in each bucket. The tradeoff to this approach is that a linear search (Big  $O(N)$ ) would be needed to find a single package if a bucket has more than one.

The distances between each address is currently stored as an adjacency matrix, but an adjacency list could also be used to store this data. This would improve the efficiency of Primm's algorithm to Big  $O(\log(N))$  but

would require adjustments to how the data is stored/gathered in the CSV file. Considering that changes to the CSV would likely be changes in how human workers operate, it may negate any speed benefits gained and introduce unneeded complications.

## Alternate Algorithms

Instead of the current approach of using Prim's algorithm and DFS to find the cycle, an alternative approach of using a Floyd-Warshall algorithm could have been used. This has potential to find a faster route, but runs slower at Big  $O(N^3)$ . Running at this speed would likely be acceptable given the current scale of operations, but would be more difficult to grow. Other alternatives would be to use a shortest path algorithm such as Dijkstra's or A\*, then use those paths to create the route.

Lastly, since the number of addresses on each truck does not currently exceed 10, a brute force approach could be used if the absolute shortest path is necessary. This would run at Big  $O(N!)$  and therefore not allow for scaling.

## Adaptability

Adaptability was a core concern when designing this program. The hash table has the ability to grow dynamically to accept new packages, the package and distance data can be loaded from different CSV files, and placing the path finding algorithm in the truck class allows the program to create more trucks as the company scales. Each step of the process is its own method, so if different data structure or algorithm is needed in the future, only the step it's replacing should need to be rewritten.

## Maintainability

Many of the steps to increase adaptability also increased maintainability. Each method is responsible for one step of the delivery process and is heavily commented. This allows future maintainers to understand the thought process that went into making this program in order to correct potential bugs or add/remove features.

## Class & Method overview

### [Package.py](#)

Method	Space-time Complexity
<code>__str__</code>	$O(1)$
<code>__eq__</code>	$O(1)$
<b>Total</b>	$2 = O(1)$

### [HashTable.py](#)

Method	Space-time Complexity
<code>count_num_addresses</code>	$O(N)$

Method	Space-time Complexity
insert_package	$O(1)$
retrieve_package	$O(1)$
handload_truck_1	$O(N)$
handload_truck_2	$O(N)$
handload_truck_3	$O(N)$
table_from_csv	$O(N)$
graph_from_csv	$O(N^2)$
update_package_nine	$O(1)$
update_package	$O(1)$
lookup_packages	$O(N^2)$
print_buckets	$O(N)$
__repr__	$O(N)$
<b>Total</b>	$2N^2 + 7N + 4 = \mathbf{O(N^2)}$

### Truck.py

Method	Space-time Complexity
find_miles_to_next	$O(1)$
deliver_package	$O(1)$
tick	$O(1)$
sort_packages	$O(N^2)$
get_packages_from_address	$O(N)$
find_minimum_spanning_tree	$O(N^2)$
get_dfs_path	$O(N^2)$
num_addresses	$O(N)$
travel	$O(1)$
__repr__	$O(1)$
<b>Total</b>	$3N^2 + 2N + 5 = \mathbf{O(N^2)}$

### Main.py

Method	Space-time Complexity
--------	-----------------------

Method	Space-time Complexity
run_deliveries	$O(N)$
deliver_packages_to_time	$O(N)$
create_new_package	$O(N)$
print_status	$O(1)$
<b>Total</b>	$3N + 1 = O(N)$

## Screenshots

First Status Check between 8:35 a.m. and 9:25 a.m.

1	195 W Oakland Ave	Salt Lake City	04115	10:30 AM	At Hub	
2	2538 S 500 E	Salt Lake City	04106	EOD	At Hub	
3	233 Canyon Rd	Salt Lake City	04103	EOD	DELIVERED at 08:56:20 on truck #2	Can only be on truck 2
4	300 W 2000 S	Salt Lake City	04115	EOD	At Hub	
5	410 S State St	Salt Lake City	04111	EOD	At Hub	
6	3060 Lester St	West Valley City	04119	10:30 AM	At Hub	Delayed on flight---will not arrive to depot until 9:05 am
7	1330 2100 S	Salt Lake City	04106	EOD	At Hub	
8	300 State St	Salt Lake City	04103	EOD	DELIVERED at 08:58:20 on truck #2	
9	300 State St	Salt Lake City	04103	EOD	At Hub	Wrong address listed
10	600 E 900 South	Salt Lake City	04105	EOD	DELIVERED at 09:07:40 on truck #2	
11	2600 Taylorsville Blvd	Salt Lake City	04118	EOD	At Hub	
12	3575 W Valley Central Station bus Loop	West Valley City	04119	EOD	At Hub	
13	2010 W 500 S	Salt Lake City	04104	10:30 AM	DELIVERED at 09:24:40 on truck #2	
14	4300 S 1300 E	Holladay	04117	10:30 AM	DELIVERED at 08:06:20 on truck #2	Must be delivered with 15, 19
15	4500 S 2300 E	Holladay	04117	9:00 AM	DELIVERED at 08:13:00 on truck #2	
16	4500 S 2300 E	Holladay	04117	10:30 AM	DELIVERED at 08:13:00 on truck #2	Must be delivered with 13, 19
17	3148 S 1100 W	Salt Lake City	04119	EOD	At Hub	
18	1488 4800 S	Salt Lake City	04123	EOD	At Hub	Can only be on truck 2
19	177 W Pryor Ave	Salt Lake City	04115	EOD	DELIVERED at 08:31:20 on truck #2	
20	3595 Main St	Salt Lake City	04115	10:30 AM	DELIVERED at 08:29:40 on truck #2	Must be delivered with 13, 15
21	3595 Main St	Salt Lake City	04115	EOD	At Hub	
22	6351 South 900 East	Murray	04121	EOD	DELIVERED at 09:23:00 on truck #1	
23	5100 South 2700 West	Salt Lake City	04118	EOD	At Hub	
24	5025 State St	Murray	04107	EOD	DELIVERED at 09:13:00 on truck #1	
25	5383 South 900 East #104	Salt Lake City	04117	10:30 AM	DELIVERED at 09:18:40 on truck #1	Delayed on flight---will not arrive to depot until 9:05 am
26	5383 South 900 East #104	Salt Lake City	04117	EOD	DELIVERED at 09:18:40 on truck #1	
27	1060 Dalton Ave S	Salt Lake City	04104	EOD	At Hub	
28	2835 Main St	Salt Lake City	04115	EOD	At Hub	Delayed on flight---will not arrive to depot until 9:05 am
29	1330 2100 S	Salt Lake City	04106	10:30 AM	At Hub	
30	300 State St	Salt Lake City	04103	10:30 AM	DELIVERED at 08:58:20 on truck #2	
31	3365 S 900 W	Salt Lake City	04119	10:30 AM	At Hub	
32	3365 S 900 W	Salt Lake City	04119	EOD	At Hub	Delayed on flight---will not arrive to depot until 9:05 am
33	2530 S 500 E	Salt Lake City	04106	EOD	At Hub	
34	4500 S 2300 E	Holladay	04117	10:30 AM	DELIVERED at 08:13:00 on truck #2	
35	1060 Dalton Ave S	Salt Lake City	04104	EOD	At Hub	
36	2300 Parkway Blvd	West Valley City	04119	EOD	At Hub	Can only be on truck 2
37	410 S State St	Salt Lake City	04111	10:30 AM	DELIVERED at 08:53:00 on truck #2	
38	410 S State St	Salt Lake City	04111	EOD	DELIVERED at 08:53:00 on truck #2	Can only be on truck 2
39	2010 W 500 S	Salt Lake City	04104	EOD	DELIVERED at 09:24:40 on truck #2	
40	300 W 2000 S	Salt Lake City	04115	10:30 AM	At Hub	
Truck #1 -- Traveling to location 9						
Package count = 12						
Mileage = 6.0						
Miles to next = 11.4						
Time = 09:25:00						
Truck #2 -- Traveling to location 7						
Package count = 2						
Mileage = 25.5						
Miles to next = 29.4						
Time = 09:25:00						
Truck #3 -- AT HUB, START OF DAY						
Package count = 8						
Mileage = 0.0						
Miles to next = 2.0						
Total miles driven = 31.5						
Current time = 09:25:00						

Second Status Check between 8:35 a.m. and 9:25 a.m.

1	195 W Oakland Ave	Salt Lake City	84115	10:30 AM	DELIVERED at 09:49:20 on truck #1	
2	2530 S 500 E	Salt Lake City	84106	EOO	DELIVERED at 09:43:00 on truck #1	
3	233 Canyon Rd	Salt Lake City	84103	EOO	DELIVERED at 08:56:20 on truck #2	Can only be on truck 2
4	380 W 2880 S	Salt Lake City	84115	EOO	DELIVERED at 09:53:00 on truck #1	
5	410 S State St	Salt Lake City	84111	EOO	At Hub	
6	3000 Lester St	West Valley City	84119	10:30 AM	DELIVERED at 10:05:00 on truck #1	Delayed on flight---will not arrive to depot until 9:05 am
7	1330 2100 S	Salt Lake City	84106	EOO	At Hub	
8	300 State St	Salt Lake City	84103	EOO	DELIVERED at 08:58:20 on truck #2	
9	410 S State St	Salt Lake City	84103	EOO	At Hub	Wrong address listed
10	600 E 900 South	Salt Lake City	84105	EOO	DELIVERED at 09:07:40 on truck #2	
11	2600 Taylorsville Blvd	Salt Lake City	84118	EOO	At Hub	
12	3575 W Valley Central Station bus Loop	West Valley City	84119	EOO	At Hub	
13	2010 W 500 S	Salt Lake City	84104	10:30 AM	DELIVERED at 09:24:40 on truck #2	
14	4300 S 1300 E	Millcreek	84117	10:30 AM	DELIVERED at 08:06:20 on truck #2	Must be delivered with 15, 19
15	4500 S 2300 E	Holladay	84117	9:00 AM	DELIVERED at 08:13:00 on truck #2	
16	4500 S 2300 E	Holladay	84117	10:30 AM	DELIVERED at 08:13:00 on truck #2	Must be delivered with 11, 19
17	3140 S 1100 W	Salt Lake City	84119	EOO	DELIVERED at 10:00:40 on truck #1	
18	1400 4000 S	Salt Lake City	84123	EOO	DELIVERED at 09:51:20 on truck #2	Can only be on truck 2
19	177 W Price Ave	Salt Lake City	84115	EOO	DELIVERED at 08:51:20 on truck #2	
20	3595 Main St	Salt Lake City	84115	10:30 AM	DELIVERED at 08:29:40 on truck #2	Must be delivered with 13, 15
21	3595 Main St	Salt Lake City	84115	EOO	At Hub	
22	6351 South 900 East	Murray	84121	EOO	DELIVERED at 09:23:00 on truck #1	
23	5100 South 2700 West	Salt Lake City	84118	EOO	At Hub	
24	5025 State St	Murray	84107	EOO	DELIVERED at 09:13:00 on truck #1	
25	5303 South 900 East #104	Salt Lake City	84117	10:30 AM	DELIVERED at 09:18:40 on truck #1	Delayed on flight---will not arrive to depot until 9:05 am
26	5303 South 900 East #104	Salt Lake City	84117	EOO	DELIVERED at 09:18:40 on truck #1	
27	1000 Dalton Ave S	Salt Lake City	84104	EOO	At Hub	
28	2835 Main St	Salt Lake City	84115	EOO	DELIVERED at 09:46:40 on truck #1	Delayed on flight---will not arrive to depot until 9:05 am
29	1330 2100 S	Salt Lake City	84106	10:30 AM	At Hub	
30	300 State St	Salt Lake City	84103	10:30 AM	DELIVERED at 08:58:20 on truck #2	
31	3365 S 900 W	Salt Lake City	84119	10:30 AM	DELIVERED at 09:58:40 on truck #1	
32	3365 S 900 W	Salt Lake City	84119	EOO	DELIVERED at 09:58:40 on truck #1	Delayed on flight---will not arrive to depot until 9:05 am
33	2530 S 500 E	Salt Lake City	84106	EOO	DELIVERED at 09:43:00 on truck #1	
34	4500 S 2300 E	Holladay	84117	10:30 AM	DELIVERED at 08:13:00 on truck #2	
35	1000 Dalton Ave S	Salt Lake City	84104	EOO	At Hub	
36	2300 Parkway Blvd	West Valley City	84119	EOO	DELIVERED at 09:38:00 on truck #2	Can only be on truck 2
37	410 S State St	Salt Lake City	84111	10:30 AM	DELIVERED at 08:53:00 on truck #2	
38	410 S State St	Salt Lake City	84111	EOO	DELIVERED at 08:53:00 on truck #2	Can only be on truck 2
39	2010 W 500 S	Salt Lake City	84104	EOO	DELIVERED at 09:24:40 on truck #2	
40	380 W 2880 S	Salt Lake City	84115	10:30 AM	DELIVERED at 09:53:00 on truck #1	
Truck #1 -- Travelling to location 2						
Package count = 2						
Mileage = 24.0						
Miles to next = 24.5						
Time = 10:25:00						
Truck #2 -- Returning to hub						
Package count = 0						
Mileage = 43.5						
Miles to next = 44.4						
Time = 10:25:00						
Truck #3 -- AT HUB, START OF DAY						
Package count = 0						
Mileage = 0.0						
Miles to next = 2.0						
Total miles driven = 67.5						
Current time = 10:25:00						

### Third Status Check between 12:03 p.m. and 1:12 p.m.

ID	ADDRESS	CITY	ZIP	DEADLINE	STATUS	INSTRUCTIONS
1	195 W Oakland Ave	Salt Lake City	84115	10:30 AM	DELIVERED at 09:49:20 on truck #1	
2	2530 S 500 E	Salt Lake City	84106	EOO	DELIVERED at 09:43:00 on truck #1	
3	233 Canyon Rd	Salt Lake City	84103	EOO	DELIVERED at 08:56:20 on truck #2	Can only be on truck 2
4	380 W 2880 S	Salt Lake City	84115	EOO	DELIVERED at 09:53:00 on truck #1	
5	410 S State St	Salt Lake City	84111	EOO	DELIVERED at 08:51:20 on truck #2	
6	3000 Lester St	West Valley City	84119	10:30 AM	DELIVERED at 10:05:00 on truck #1	Delayed on flight---will not arrive to depot until 9:05 am
7	1330 2100 S	Salt Lake City	84106	EOO	DELIVERED at 10:26:40 on truck #1	
8	300 State St	Salt Lake City	84103	EOO	DELIVERED at 08:58:20 on truck #2	
9	410 S State St	Salt Lake City	84103	EOO	DELIVERED at 10:51:20 on truck #3	Wrong address listed
10	600 E 900 South	Salt Lake City	84105	EOO	DELIVERED at 09:07:40 on truck #2	
11	2600 Taylorsville Blvd	Salt Lake City	84118	EOO	DELIVERED at 11:31:40 on truck #3	
12	3575 W Valley Central Station bus Loop	West Valley City	84119	EOO	DELIVERED at 11:58:40 on truck #3	
13	2010 W 500 S	Salt Lake City	84104	10:30 AM	DELIVERED at 09:24:40 on truck #2	
14	4300 S 1300 E	Millcreek	84117	10:30 AM	DELIVERED at 08:06:20 on truck #2	Must be delivered with 15, 19
15	4500 S 2300 E	Holladay	84117	9:00 AM	DELIVERED at 08:13:00 on truck #2	
16	4500 S 2300 E	Holladay	84117	10:30 AM	DELIVERED at 08:13:00 on truck #2	Must be delivered with 13, 19
17	3140 S 1100 W	Salt Lake City	84119	EOO	DELIVERED at 10:00:40 on truck #1	
18	1400 4000 S	Salt Lake City	84123	EOO	DELIVERED at 09:51:20 on truck #2	Can only be on truck 2
19	177 W Price Ave	Salt Lake City	84115	EOO	DELIVERED at 08:51:20 on truck #2	
20	3595 Main St	Salt Lake City	84115	10:30 AM	DELIVERED at 08:29:40 on truck #2	Must be delivered with 13, 15
21	3595 Main St	Salt Lake City	84115	EOO	DELIVERED at 10:35:00 on truck #3	
22	6351 South 900 East	Murray	84121	EOO	DELIVERED at 09:23:00 on truck #1	
23	5100 South 2700 West	Salt Lake City	84118	EOO	DELIVERED at 11:30:20 on truck #3	
24	5025 State St	Murray	84107	EOO	DELIVERED at 09:13:00 on truck #1	
25	5303 South 900 East #104	Salt Lake City	84117	10:30 AM	DELIVERED at 09:18:40 on truck #1	Delayed on flight---will not arrive to depot until 9:05 am
26	5303 South 900 East #104	Salt Lake City	84117	EOO	DELIVERED at 09:18:40 on truck #1	
27	1000 Dalton Ave S	Salt Lake City	84104	EOO	DELIVERED at 11:07:20 on truck #3	
28	2835 Main St	Salt Lake City	84115	EOO	DELIVERED at 09:46:40 on truck #1	Delayed on flight---will not arrive to depot until 9:05 am
29	1330 2100 S	Salt Lake City	84106	10:30 AM	DELIVERED at 10:26:40 on truck #1	
30	300 State St	Salt Lake City	84103	10:30 AM	DELIVERED at 08:58:20 on truck #2	
31	3365 S 900 W	Salt Lake City	84119	10:30 AM	DELIVERED at 09:58:40 on truck #1	
32	3365 S 900 W	Salt Lake City	84119	EOO	DELIVERED at 09:58:40 on truck #1	Delayed on flight---will not arrive to depot until 9:05 am
33	2530 S 500 E	Salt Lake City	84106	EOO	DELIVERED at 09:43:00 on truck #1	
34	4500 S 2300 E	Holladay	84117	10:30 AM	DELIVERED at 08:13:00 on truck #2	
35	1000 Dalton Ave S	Salt Lake City	84104	EOO	DELIVERED at 11:07:20 on truck #3	
36	2300 Parkway Blvd	West Valley City	84119	EOO	DELIVERED at 09:38:00 on truck #2	Can only be on truck 2
37	410 S State St	Salt Lake City	84111	10:30 AM	DELIVERED at 08:53:00 on truck #2	
38	410 S State St	Salt Lake City	84111	EOO	DELIVERED at 08:53:00 on truck #2	Can only be on truck 2
39	2010 W 500 S	Salt Lake City	84104	EOO	DELIVERED at 09:24:40 on truck #2	
40	380 W 2880 S	Salt Lake City	84115	10:30 AM	DELIVERED at 09:53:00 on truck #1	
Truck #1 -- Deliveries complete						
Package count = 0						
Mileage = 28.3						
Time = 10:30:20						
Truck #2 -- Deliveries complete						
Package count = 0						
Mileage = 44.4						
Time = 10:28:00						
Truck #3 -- Deliveries complete						
Package count = 0						
Mileage = 34.7						
Time = 12:24:00						
Total miles driven = 107.4						
Current time = 13:12:00						

### Code Execution - Main Screen





```

Current time = 11:12

1) Set time of day
2) Print current package & truck status
3) Insert new package
4) Lookup package based on address
5) Lookup package based on city
6) Lookup package based on zip code
7) Lookup package based on weight
8) Lookup package based on deadline
9) Lookup package based on status
0) Exit program

Please select an option: 5
Enter the city you would like to lookup: Salt
Packages with City: Salt
-----
ID = 01 Address = [05] 195 N Oakland Ave Salt Lake City,UT 84115 Weight = 21 Instruction = Status = DELIVERED at 09:49:20 on truck #1
ID = 02 Address = [09] 2530 S 900 E Salt Lake City,UT 84106 Weight = 44 Instruction = Status = DELIVERED at 09:43:00 on truck #1
ID = 03 Address = [08] 233 Canyon Rd Salt Lake City,UT 84103 Weight = 2 Instruction = Can only be on truck 2 Status = DELIVERED at 08:56:20 on truck #2
ID = 04 Address = [10] 200 W 2000 S Salt Lake City,UT 84115 Weight = 4 Instruction = Status = DELIVERED at 08:53:00 on truck #1
ID = 05 Address = [10] 419 S State St Salt Lake City,UT 84111 Weight = 5 Instruction = Status = DELIVERED at 08:53:00 on truck #3
ID = 07 Address = [02] 1330 2100 S Salt Lake City,UT 84106 Weight = 8 Instruction = Status = DELIVERED at 10:26:40 on truck #1
ID = 08 Address = [12] 300 State St Salt Lake City,UT 84103 Weight = 9 Instruction = Status = DELIVERED at 08:50:20 on truck #2
ID = 09 Address = [10] 419 S State St Salt Lake City,UT 84103 Weight = 2 Instruction = Wrong address listed Status = DELIVERED at 09:53:00 on truck #3
ID = 10 Address = [25] 600 E 900 South Salt Lake City,UT 84105 Weight = 1 Instruction = Status = DELIVERED at 09:07:40 on truck #2
ID = 11 Address = [10] 2600 Taylorsville Blvd Salt Lake City,UT 84118 Weight = 1 Instruction = Status = DELIVERED at 11:31:40 on truck #3
ID = 13 Address = [06] 2010 W 900 S Salt Lake City,UT 84104 Weight = 2 Instruction = Status = DELIVERED at 09:24:40 on truck #2
ID = 17 Address = [14] 3148 S 1100 W Salt Lake City,UT 84119 Weight = 2 Instruction = Status = DELIVERED at 10:00:40 on truck #1
ID = 18 Address = [03] 3400 4000 S Salt Lake City,UT 84123 Weight = 6 Instruction = Can only be on truck 2 Status = DELIVERED at 09:51:20 on truck #2
ID = 19 Address = [04] 177 W Prince Ave Salt Lake City,UT 84115 Weight = 17 Instruction = Must be delivered with 13, 15 Status = DELIVERED at 08:31:20 on truck #2
ID = 20 Address = [17] 3995 Main St Salt Lake City,UT 84115 Weight = 37 Instruction = Status = DELIVERED at 08:29:40 on truck #2
ID = 21 Address = [17] 3995 Main St Salt Lake City,UT 84115 Weight = 3 Instruction = Status = DELIVERED at 10:35:00 on truck #3
ID = 23 Address = [21] 5100 South 7000 West Salt Lake City,UT 84118 Weight = 5 Instruction = Status = DELIVERED at 11:30:20 on truck #3
ID = 24 Address = [24] 5383 South 900 East #104 Salt Lake City,UT 84117 Weight = 7 Instruction = Delayed on flight---will not arrive to depot until 9:05 am Status = DELIVERED at 09:18:40 on truck #1
ID = 26 Address = [24] 5383 South 900 East #104 Salt Lake City,UT 84117 Weight = 25 Instruction = Status = DELIVERED at 09:18:40 on truck #1
ID = 27 Address = [01] 1002 Dalton Ave S Salt Lake City,UT 84104 Weight = 5 Instruction = Status = DELIVERED at 11:07:20 on truck #3
ID = 28 Address = [11] 2835 Main St Salt Lake City,UT 84115 Weight = 7 Instruction = Delayed on flight---will not arrive to depot until 9:05 am Status = DELIVERED at 09:46:40 on truck #1
ID = 29 Address = [02] 1330 2100 S Salt Lake City,UT 84106 Weight = 2 Instruction = Status = DELIVERED at 10:26:40 on truck #1
ID = 30 Address = [12] 300 State St Salt Lake City,UT 84103 Weight = 1 Instruction = Status = DELIVERED at 08:50:20 on truck #2
ID = 31 Address = [15] 3365 S 900 W Salt Lake City,UT 84119 Weight = 1 Instruction = Status = DELIVERED at 09:58:40 on truck #1
ID = 32 Address = [15] 3365 S 900 W Salt Lake City,UT 84119 Weight = 1 Instruction = Delayed on flight---will not arrive to depot until 9:05 am Status = DELIVERED at 09:58:40 on truck #1
ID = 33 Address = [09] 2530 S 900 E Salt Lake City,UT 84106 Weight = 1 Instruction = Status = DELIVERED at 09:43:00 on truck #1
ID = 35 Address = [01] 1060 Dalton Ave S Salt Lake City,UT 84104 Weight = 88 Instruction = Status = DELIVERED at 11:07:20 on truck #3
ID = 37 Address = [10] 419 S State St Salt Lake City,UT 84111 Weight = 2 Instruction = Status = DELIVERED at 08:53:00 on truck #2
ID = 38 Address = [10] 419 S State St Salt Lake City,UT 84111 Weight = 9 Instruction = Can only be on truck 2 Status = DELIVERED at 08:53:00 on truck #2
ID = 39 Address = [08] 2010 W 900 S Salt Lake City,UT 84104 Weight = 9 Instruction = Status = DELIVERED at 09:24:40 on truck #2
ID = 40 Address = [10] 300 W 2000 S Salt Lake City,UT 84115 Weight = 45 Instruction = Status = DELIVERED at 09:53:00 on truck #1

```

## Sources

The main resource used during program creation was the textbook provided by WGU and Zybooks.

Learn.zybooks.com. (n.d.). zyBooks. [online] Available at: <https://learn.zybooks.com/zybook/WGUC950AY20182019/>