Core Algorithm Overview

Section 1

A The algorithm used is a Greedy implementation where the next path taken is the immediate shortest path from the current node. The current node is marked as visited and removed from the pool of nodes able to be visited in the future.

B1 This is pseudocode for checking package status of one or many packages at an input time.

```
Package Delivery function (params: Truck, package ID list, Time)
      If Truck's clock time exceeds Time
            Stop function

      For each package ID in list
            Get Package from package hash table via package ID
            Get location ID from location hash table via Package
            Add Package and location ID (destination) to Truck

      Greedy function (param: current node)
            Distance = a large number
            Remove current node from Truck destinations

            While Truck's clock time is less than Time
                  For every other node in Truck destinations
                        If the path to this node is shorter than Distance
                              This path becomes the new Distance
                              This node becomes the Next Node to visit

                  If a path was found
                        Add distance travel time to Truck's clock
                        Call Greedy on the Next Node
                  Else
                        Distance becomes path to origin node
                        Add distance travel time to Truck's clock
```

Once this function ends, a loop in an outer scope prints each package's status at the time requested.

The pseudocode for viewing truck mileage and routing is similar. Aside from lacking checks on time because the route is being viewed in its totality, there is a print statement towards the end of every Greedy iteration via a helper function.

```
Greedy Helper function (params: Distance, Node, Next Node)
      Add distance travel time to Truck's clock
      Add distance traveled to Truck's mileage
      Print the time, the distance traveled, and the nodes traversed
```

B2 The program is written in Python 3.8 and runs on a local machine. It is executed on the PyCharm IDE, though other IDEs may be just as suitable. CSV files manually added to the project have their data extracted and manipulated by the program. The program is controlled by the user through the IDE's console.

B3 The truck routing algorithm has a list of n delivery locations. Each function call finds the next location to visit from the current location. Each location is visited once and only once, so the function call is made n times. Each iteration removes the current location from the pool of locations, runs a loop through the remaining locations, and chooses one to visit. This is repeated until there are no more locations to visit and the truck returns to its original location. This resembles the arithmetic sequence:

$$n + (n-1) + (n-2) + \cdots + 2 + 1 = \frac{n(n+1)}{2}$$

Thus, the algorithm has a time complexity of $O(n^2)$. The best and worst growth rate is the same, so the algorithm is tightly bound as well, $\Theta(n^2)$. Other time complexity is noted throughout the program in comments.

Regarding space complexity, a graph or similar data structure is not used because retrieval and comparisons are made via hash table lookup. Each truck uses a dictionary of sets where each location-key has one or multiple package-values. Overall, space complexity is O(n).

B4 I would not recommend my program for general use because, frankly, the program is bespoke. That said, the fundamental algorithm is sound. The Greedy algorithm could handle thousands, if not tens of thousands of packages just fine. However, using another algorithm like Dijkstra's or 2-opt would be a better business decision because those are more likely to find shorter routes. Additionally, to make the program market-viable, significant changes would have to be made to automatically process special notes and conditions, and automate the process of sorting packages for arrival, departure, and other delivery constraints, as well as fixing other issues like programmatically finding hash table sizes with good load factors instead of hard coding them in.

B5 The program's limiting factor regarding time complexity is the algorithm. The Greedy algorithm runs in polynomial time and is efficient for our purposes. As far as maintainability, the algorithm or other data structures could be re-written without affecting other parts of the program. I used native Python methods when possible instead of writing my own, which should improve maintainability if the project is handed off.

B6 The main data structure in use is the hash table I created, which is simply a list of lists. The inner lists act as the buckets for the key-value pairs. It is about as simple as can be, where collision is resolved via separate chaining. (However, in the instance I used a unique integer as the key, its "hash" is the integer itself, and the hash table became free of collisions and effectively turned into a direct access table.) Insertion and retrieval are O(1) on average.

G The following three screenshots display package status at 8:59, 9:48, and 12:05 respectively.

H The following fourth and fifth screenshots display code execution of the truck routes and mileage and single package status view.

•■• ALL PACKAGES AT 08:59:37 •■•

ID	Address	City	State	Zip	Deadline	Kilos	Status	Note
1	195 W Oakland Ave	Salt Lake City	UT	84115	10:30:00	21	Delivered	
2	2530 S 500 E	Salt Lake City	UT	84106	17:00:00	44	At Facility	
3	233 Canyon Rd	Salt Lake City	UT	84103	17:00:00	2	At Facility	Can only be on truck 2
4	380 W 2880 S	Salt Lake City	UT	84115	17:00:00	4	At Facility	
5	410 S State St	Salt Lake City	UT	84111	17:00:00	5	At Facility	
6	3060 Lester St	West Valley City	UT	84119	10:30:00	88	At Facility	Delayed on flight---will not arrive to depot until 9:05
7	1330 2100 S	Salt Lake City	UT	84106	17:00:00	8	At Facility	
8	300 State St	Salt Lake City	UT	84103	17:00:00	9	At Facility	
9	410 S State St	Salt Lake City	UT	84111	17:00:00	2	At Facility	Wrong address listed
10	600 E 900 South	Salt Lake City	UT	84105	17:00:00	1	At Facility	
11	2600 Taylorsville Blvd	Salt Lake City	UT	84118	17:00:00	1	At Facility	
12	3575 W Valley Central Station Bus Loop	West Valley City	UT	84119	17:00:00	1	At Facility	
13	2010 W 500 S	Salt Lake City	UT	84104	10:30:00	2	Out for Delivery	
14	4300 S 1300 E	Millcreek	UT	84117	10:30:00	88	Delivered	Must be delivered with 15, 19
15	4580 S 2300 E	Holladay	UT	84117	09:00:00	4	Delivered	
16	4580 S 2300 E	Holladay	UT	84117	10:30:00	88	Delivered	Must be delivered with 13, 19
17	3148 S 1100 W	Salt Lake City	UT	84119	17:00:00	2	At Facility	
18	1488 4800 S	Salt Lake City	UT	84123	17:00:00	6	At Facility	Can only be on truck 2
19	177 W Price Ave	Salt Lake City	UT	84115	17:00:00	37	Delivered	
20	3595 Main St	Salt Lake City	UT	84115	10:30:00	37	Delivered	Must be delivered with 13, 15
21	3595 Main St	Salt Lake City	UT	84115	17:00:00	3	At Facility	
22	6351 South 900 East	Murray	UT	84121	17:00:00	2	At Facility	
23	5100 South 2700 West	Salt Lake City	UT	84118	17:00:00	5	At Facility	
24	5025 State St	Murray	UT	84107	17:00:00	7	At Facility	
25	5383 South 900 East #104	Salt Lake City	UT	84117	10:30:00	7	At Facility	Delayed on flight---will not arrive to depot until 9:05
26	5383 South 900 East #104	Salt Lake City	UT	84117	17:00:00	25	At Facility	
27	1060 Dalton Ave S	Salt Lake City	UT	84104	17:00:00	5	At Facility	
28	2835 Main St	Salt Lake City	UT	84115	17:00:00	7	At Facility	Delayed on flight---will not arrive to depot until 9:05
29	1330 2100 S	Salt Lake City	UT	84106	10:30:00	2	Delivered	
30	300 State St	Salt Lake City	UT	84103	10:30:00	1	Out for Delivery	
31	3365 S 900 W	Salt Lake City	UT	84119	10:30:00	1	Out for Delivery	
32	3365 S 900 W	Salt Lake City	UT	84119	17:00:00	1	At Facility	Delayed on flight---will not arrive to depot until 9:05
33	2530 S 500 E	Salt Lake City	UT	84106	17:00:00	1	At Facility	
34	4580 S 2300 E	Holladay	UT	84117	10:30:00	2	Delivered	
35	1060 Dalton Ave S	Salt Lake City	UT	84104	17:00:00	88	At Facility	
36	2300 Parkway Blvd	West Valley City	UT	84119	17:00:00	88	At Facility	Can only be on truck 2
37	410 S State St	Salt Lake City	UT	84111	10:30:00	2	Delivered	
38	410 S State St	Salt Lake City	UT	84111	17:00:00	9	At Facility	Can only be on truck 2
39	2010 W 500 S	Salt Lake City	UT	84104	17:00:00	9	At Facility	
40	380 W 2880 S	Salt Lake City	UT	84115	10:30:00	45	Delivered	

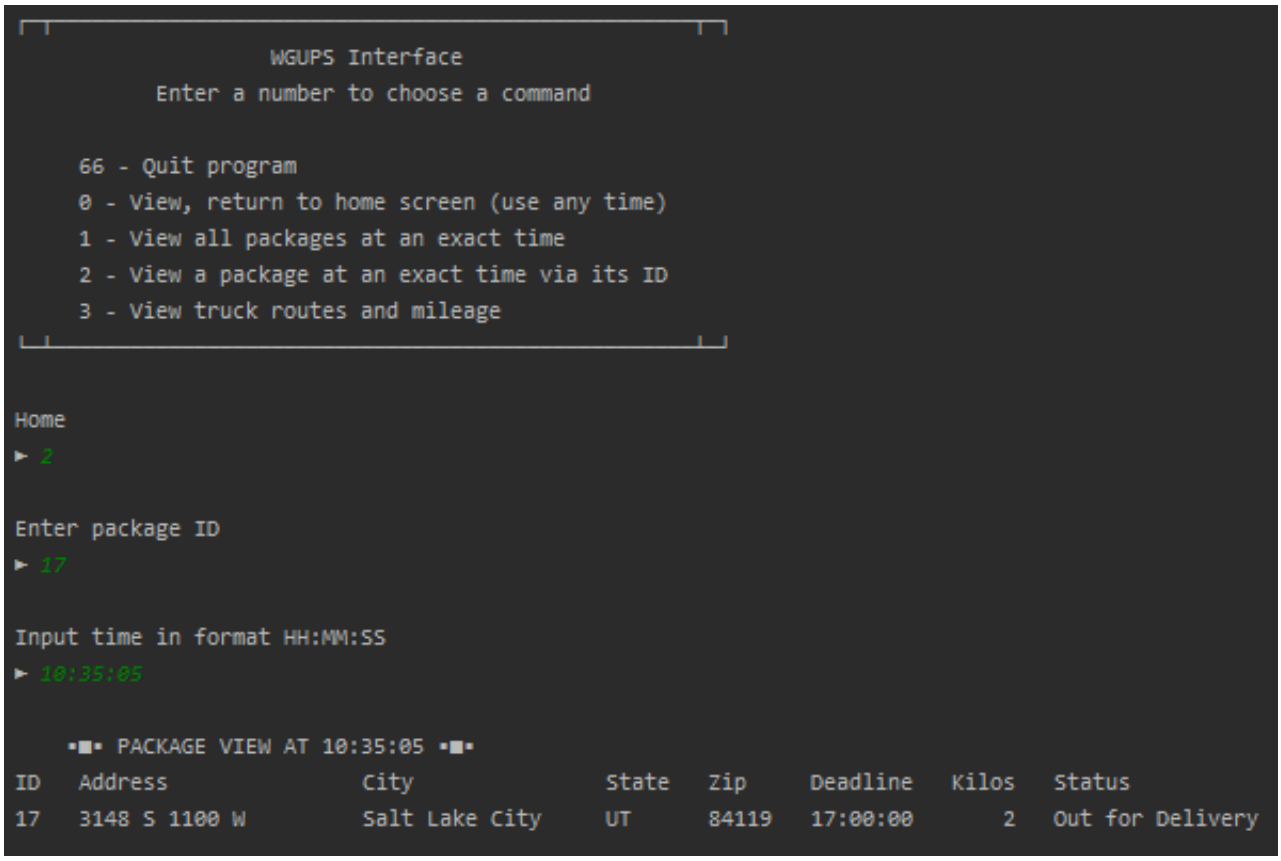
▪■▪ ALL PACKAGES AT 09:48:00 ▪■▪

ID	Address	City	State	Zip	Deadline	Kilos	Status	Note
1	195 W Oakland Ave	Salt Lake City	UT	84115	10:30:00	21	Delivered	
2	2530 S 500 E	Salt Lake City	UT	84106	17:00:00	44	Delivered	
3	233 Canyon Rd	Salt Lake City	UT	84103	17:00:00	2	Out for Delivery	Can only be on truck 2
4	380 W 2880 S	Salt Lake City	UT	84115	17:00:00	4	Delivered	
5	410 S State St	Salt Lake City	UT	84111	17:00:00	5	Out for Delivery	
6	3060 Lester St	West Valley City	UT	84119	10:30:00	88	Delivered	Delayed on flight---will not arrive to depot until 9:05
7	1330 2100 S	Salt Lake City	UT	84106	17:00:00	8	Out for Delivery	
8	300 State St	Salt Lake City	UT	84103	17:00:00	9	Out for Delivery	
9	410 S State St	Salt Lake City	UT	84111	17:00:00	2	At Facility	Wrong address listed
10	600 E 900 South	Salt Lake City	UT	84105	17:00:00	1	At Facility	
11	2600 Taylorsville Blvd	Salt Lake City	UT	84118	17:00:00	1	At Facility	
12	3575 W Valley Central Station Bus Loop	West Valley City	UT	84119	17:00:00	1	At Facility	
13	2010 W 500 S	Salt Lake City	UT	84104	10:30:00	2	Delivered	
14	4300 S 1300 E	Millcreek	UT	84117	10:30:00	88	Delivered	Must be delivered with 15, 19
15	4580 S 2300 E	Holladay	UT	84117	09:00:00	4	Delivered	
16	4580 S 2300 E	Holladay	UT	84117	10:30:00	88	Delivered	Must be delivered with 13, 19
17	3148 S 1100 W	Salt Lake City	UT	84119	17:00:00	2	At Facility	
18	1488 4800 S	Salt Lake City	UT	84123	17:00:00	6	Out for Delivery	Can only be on truck 2
19	177 W Price Ave	Salt Lake City	UT	84115	17:00:00	37	Delivered	
20	3595 Main St	Salt Lake City	UT	84115	10:30:00	37	Delivered	Must be delivered with 13, 15
21	3595 Main St	Salt Lake City	UT	84115	17:00:00	3	At Facility	
22	6351 South 900 East	Murray	UT	84121	17:00:00	2	At Facility	
23	5100 South 2700 West	Salt Lake City	UT	84118	17:00:00	5	At Facility	
24	5025 State St	Murray	UT	84107	17:00:00	7	At Facility	
25	5383 South 900 East #104	Salt Lake City	UT	84117	10:30:00	7	Delivered	Delayed on flight---will not arrive to depot until 9:05
26	5383 South 900 East #104	Salt Lake City	UT	84117	17:00:00	25	At Facility	
27	1060 Dalton Ave S	Salt Lake City	UT	84104	17:00:00	5	At Facility	
28	2835 Main St	Salt Lake City	UT	84115	17:00:00	7	Delivered	Delayed on flight---will not arrive to depot until 9:05
29	1330 2100 S	Salt Lake City	UT	84106	10:30:00	2	Delivered	
30	300 State St	Salt Lake City	UT	84103	10:30:00	1	Delivered	
31	3365 S 900 W	Salt Lake City	UT	84119	10:30:00	1	Delivered	
32	3365 S 900 W	Salt Lake City	UT	84119	17:00:00	1	Delivered	Delayed on flight---will not arrive to depot until 9:05
33	2530 S 500 E	Salt Lake City	UT	84106	17:00:00	1	At Facility	
34	4580 S 2300 E	Holladay	UT	84117	10:30:00	2	Delivered	
35	1060 Dalton Ave S	Salt Lake City	UT	84104	17:00:00	88	At Facility	
36	2300 Parkway Blvd	West Valley City	UT	84119	17:00:00	88	Delivered	Can only be on truck 2
37	410 S State St	Salt Lake City	UT	84111	10:30:00	2	Delivered	
38	410 S State St	Salt Lake City	UT	84111	17:00:00	9	Out for Delivery	Can only be on truck 2
39	2010 W 500 S	Salt Lake City	UT	84104	17:00:00	9	At Facility	
40	380 W 2880 S	Salt Lake City	UT	84115	10:30:00	45	Delivered	

▪■▪ ALL PACKAGES AT 12:05:00 ▪■▪

ID	Address	City	State	Zip	Deadline	Kilos	Status	Note
1	195 W Oakland Ave	Salt Lake City	UT	84115	10:30:00	21	Delivered	
2	2530 S 500 E	Salt Lake City	UT	84106	17:00:00	44	Delivered	
3	233 Canyon Rd	Salt Lake City	UT	84103	17:00:00	2	Delivered	Can only be on truck 2
4	380 W 2880 S	Salt Lake City	UT	84115	17:00:00	4	Delivered	
5	410 S State St	Salt Lake City	UT	84111	17:00:00	5	Delivered	
6	3060 Lester St	West Valley City	UT	84119	10:30:00	88	Delivered	Delayed on flight---will not arrive to depot until 9:05
7	1330 2100 S	Salt Lake City	UT	84106	17:00:00	8	Delivered	
8	300 State St	Salt Lake City	UT	84103	17:00:00	9	Delivered	
9	410 S State St	Salt Lake City	UT	84111	17:00:00	2	Delivered	Wrong address listed
10	600 E 900 South	Salt Lake City	UT	84105	17:00:00	1	Delivered	
11	2600 Taylorsville Blvd	Salt Lake City	UT	84118	17:00:00	1	Delivered	
12	3575 W Valley Central Station Bus Loop	West Valley City	UT	84119	17:00:00	1	Delivered	
13	2010 W 500 S	Salt Lake City	UT	84104	10:30:00	2	Delivered	
14	4300 S 1300 E	Millcreek	UT	84117	10:30:00	88	Delivered	Must be delivered with 15, 19
15	4580 S 2300 E	Holladay	UT	84117	09:00:00	4	Delivered	
16	4580 S 2300 E	Holladay	UT	84117	10:30:00	88	Delivered	Must be delivered with 13, 19
17	3148 S 1100 W	Salt Lake City	UT	84119	17:00:00	2	Delivered	
18	1488 4800 S	Salt Lake City	UT	84123	17:00:00	6	Delivered	Can only be on truck 2
19	177 W Price Ave	Salt Lake City	UT	84115	17:00:00	37	Delivered	
20	3595 Main St	Salt Lake City	UT	84115	10:30:00	37	Delivered	Must be delivered with 13, 15
21	3595 Main St	Salt Lake City	UT	84115	17:00:00	3	Delivered	
22	6351 South 900 East	Murray	UT	84121	17:00:00	2	Delivered	
23	5100 South 2700 West	Salt Lake City	UT	84118	17:00:00	5	Delivered	
24	5025 State St	Murray	UT	84107	17:00:00	7	Delivered	
25	5383 South 900 East #104	Salt Lake City	UT	84117	10:30:00	7	Delivered	Delayed on flight---will not arrive to depot until 9:05
26	5383 South 900 East #104	Salt Lake City	UT	84117	17:00:00	25	Delivered	
27	1060 Dalton Ave S	Salt Lake City	UT	84104	17:00:00	5	Delivered	
28	2835 Main St	Salt Lake City	UT	84115	17:00:00	7	Delivered	Delayed on flight---will not arrive to depot until 9:05
29	1330 2100 S	Salt Lake City	UT	84106	10:30:00	2	Delivered	
30	300 State St	Salt Lake City	UT	84103	10:30:00	1	Delivered	
31	3365 S 900 W	Salt Lake City	UT	84119	10:30:00	1	Delivered	
32	3365 S 900 W	Salt Lake City	UT	84119	17:00:00	1	Delivered	Delayed on flight---will not arrive to depot until 9:05
33	2530 S 500 E	Salt Lake City	UT	84106	17:00:00	1	Delivered	
34	4580 S 2300 E	Holladay	UT	84117	10:30:00	2	Delivered	
35	1060 Dalton Ave S	Salt Lake City	UT	84104	17:00:00	88	Delivered	
36	2300 Parkway Blvd	West Valley City	UT	84119	17:00:00	88	Delivered	Can only be on truck 2
37	410 S State St	Salt Lake City	UT	84111	10:30:00	2	Delivered	
38	410 S State St	Salt Lake City	UT	84111	17:00:00	9	Delivered	Can only be on truck 2
39	2010 W 500 S	Salt Lake City	UT	84104	17:00:00	9	Delivered	
40	380 W 2880 S	Salt Lake City	UT	84115	10:30:00	45	Delivered	

```
                    WGUPS Interface
            Enter a number to choose a command

     66 - Quit program
     0 - View, return to home screen (use any time)
     1 - View all packages at an exact time
     2 - View a package at an exact time via its ID
     3 - View truck routes and mileage


Home
▶ 3

    ▪■▪ TRUCK ONE ▪■▪
Clock Time   Miles   Location ID Path
08:06:20     1.9     0       ->     20
08:13:00     3.9     20      ->     21
08:29:40     8.9     21      ->     17
08:31:20     9.4     17      ->     4
08:37:00     11.1    4       ->     18
08:40:40     12.2    18      ->     5
08:50:00     15.0    5       ->     2
09:04:20     19.3    2       ->     19
09:07:40     20.3    19      ->     12
09:21:40     24.5    12      ->     6
09:41:00     30.3    6       ->     15
09:53:20     34.0    15      ->     0

    ▪■▪ TRUCK TWO ▪■▪
Clock Time   Miles   Location ID Path
09:13:00     2.4     0       ->     24
09:29:00     7.2     24      ->     9
09:32:40     8.3     9       ->     11
09:36:00     9.3     11      ->     18
09:41:40     11.0    18      ->     15
09:46:40     12.5    15      ->     13
09:52:00     14.1    13      ->     7
10:05:20     18.1    7       ->     3
10:36:00     27.3    3       ->     2
10:50:20     31.6    2       ->     19
10:53:40     32.6    19      ->     8
10:55:40     33.2    8       ->     12
11:21:00     40.8    12      ->     0

    ▪■▪ TRUCK THREE ▪■▪
Clock Time   Miles   Location ID Path
10:06:40     2.0     0       ->     17
10:14:20     4.3     17      ->     22
10:20:00     6.0     22      ->     24
10:24:20     7.3     24      ->     26
10:44:20     13.3    26      ->     9
10:55:00     16.5    9       ->     25
11:01:00     18.3    25      ->     19
11:11:40     21.5    19      ->     6
11:17:00     23.1    6       ->     1
11:32:20     27.7    1       ->     14
11:48:20     32.5    14      ->     23
11:49:40     32.9    23      ->     10
12:16:40     41.0    10      ->     16
12:42:00     48.6    16      ->     0

🚚 ID     MILEAGE
    1        34.0
    2        40.8
    3        48.6
           123.4 TOTAL
```



Section 2

I1 The biggest strengths of this Greedy algorithm are that 1) it is a relatively simple implementation and should be easier to maintain in the future, and 2) its run time is theoretically constant because the best and worst case is the same.

I2 The algorithm meets all the requirements give in the scenario because it runs in polynomial time and found a package delivery route that meets the constraints of the scenario, namely that each package is delivered on time.

I3 One alternative algorithm to implement would be Dijkstra's shortest path algorithm. It is an improvement over a Greedy implementation because instead settling for the immediate next shortest path, Dijkstra's will prioritize a "detour" through a set of nodes if the sum of that path is shorter than taking a more direct route.

Both Greedy and Dijkstra are not very fast when n is large, though they are not the worst. But if we truly do not care about time complexity, we can fall back on the brute force approach which would compare every possible combinations of paths in O(n!). Considering this application is built for a small company with very few deliverables, and likely cannot survive on as thin a profit margin as it would like, a brute force approach might be preferred in some scenarios to guarantee the optimal route.

J If I did this project again, I would make significant changes to automatically process special notes and conditions, automate the process of sorting packages for arrival, departure, and other delivery constraints, as well as fix other issues like programmatically finding hash table sizes with good load factors instead of hard coding them in.

K1 The main data structure in use is the hash table I created, described more thoroughly in part B6. The hash table is very efficient in space complexity, which is linear, and would continue to be linear regardless of an increase in program scale. For time complexity, insertion and retrieval are O(1) on average. The only extra overhead in use is during Greedy algorithm execution where path comparisons are looked-up on the fly.

K2 A graph object with methods for edge creation and deletion would be very flexible in the long term. It would be cumbersome and almost certainly less efficient to implement multi-paths comparisons without a graph object in my algorithm. It would be pertinent to use a graph data structure if another algorithm with greater space complexity concerns is to be implemented in the future.

Another useful data structure for handling pathing would be a linked list. A linked list would create a manipulatable object that could be passed around for other uses, unlike my algorithm, which spits out a solution without creating any kind of object. Moreover, a linked list would preserve the ease of insertion and retrieval that my algorithm currently has.

L No content is quoted, paraphrased, or summarized from a source.