**Section 1:**

1. The main algorithm used for this project was a greedy algorithm. A greedy algorithm chooses the optimal option when presented with a list of options at that time. This program will sort the packages according to the level of priority using heuristics and get an unordered route. The greedy algorithm will get the nearest location from the hub in that route and then recursively call the function for each location. This will get the shortest route ensuring that the packages are delivered on time.
2. Algorithm Overview:
   * 1. The program has three main parts:

First the packages are sorted according to priority. The packages with deadlines are appended first into the delivery list, then packages with specific instructions are added and then delayed packages and then EOD delivery packages are added.

The packages are loaded into three trucks and the route is established.

The second part finds the fastest route around the city. The algorithm works in this manner:

* Find the closest location from the hub.
* Use that location to find the next closest location
* Call the function recursively for each closest location in that route.

The third part delivers the packages. The route is stored in a dictionary with the location and distance in order. The program loops through the route and then for each route loops through the delivery list to find the package for that location and then delivers the package. The mileage and time are updated.

* + 1. Sorting the route:

current location = HUB

Establish route from delivery list

FunctionGetNextLocation(current location)

Get neighbors of location:

If neighbor is in established route and closest to location

Add neighbor to sorted route

Current location = neighbor

If all packages are delivered, add HUB to route

Else FunctionGetNextLocation(current location)

Deliver Packages:

For location, in route:

Update time

Update mileage

For package in deliverylist:

If package goes to location:

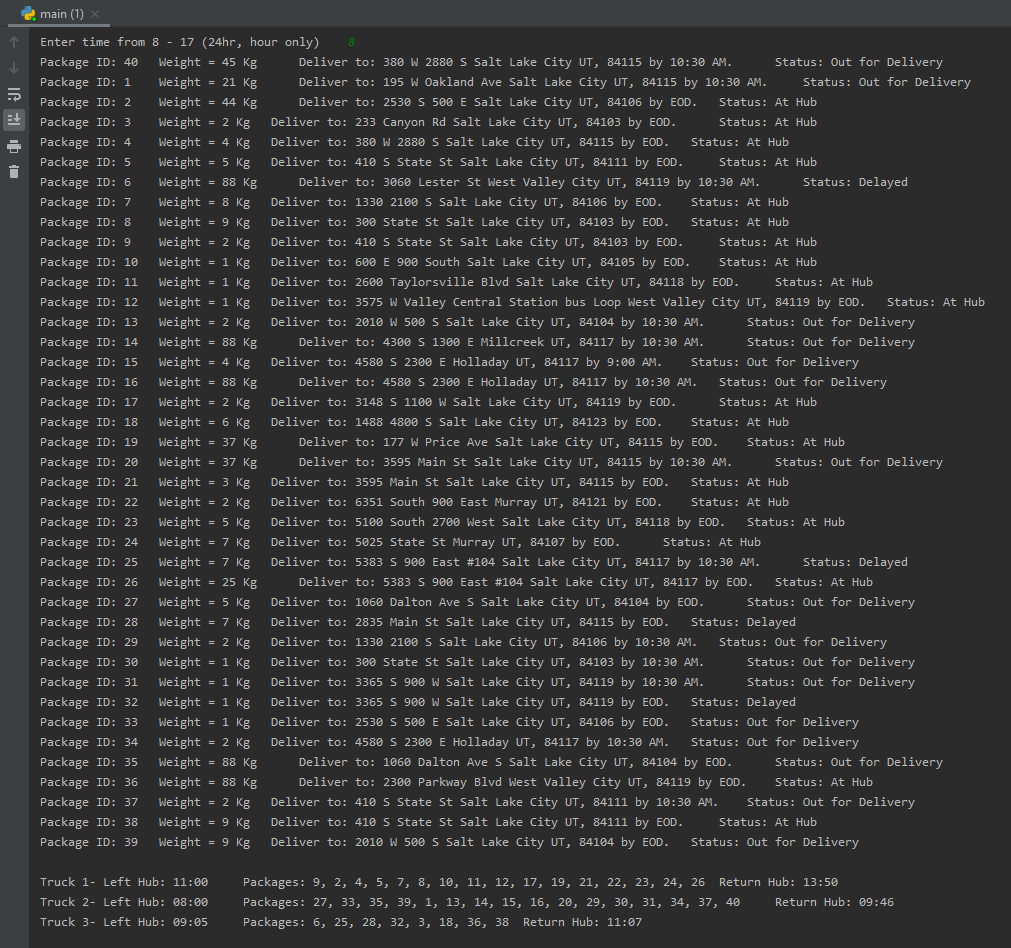
Deliver package

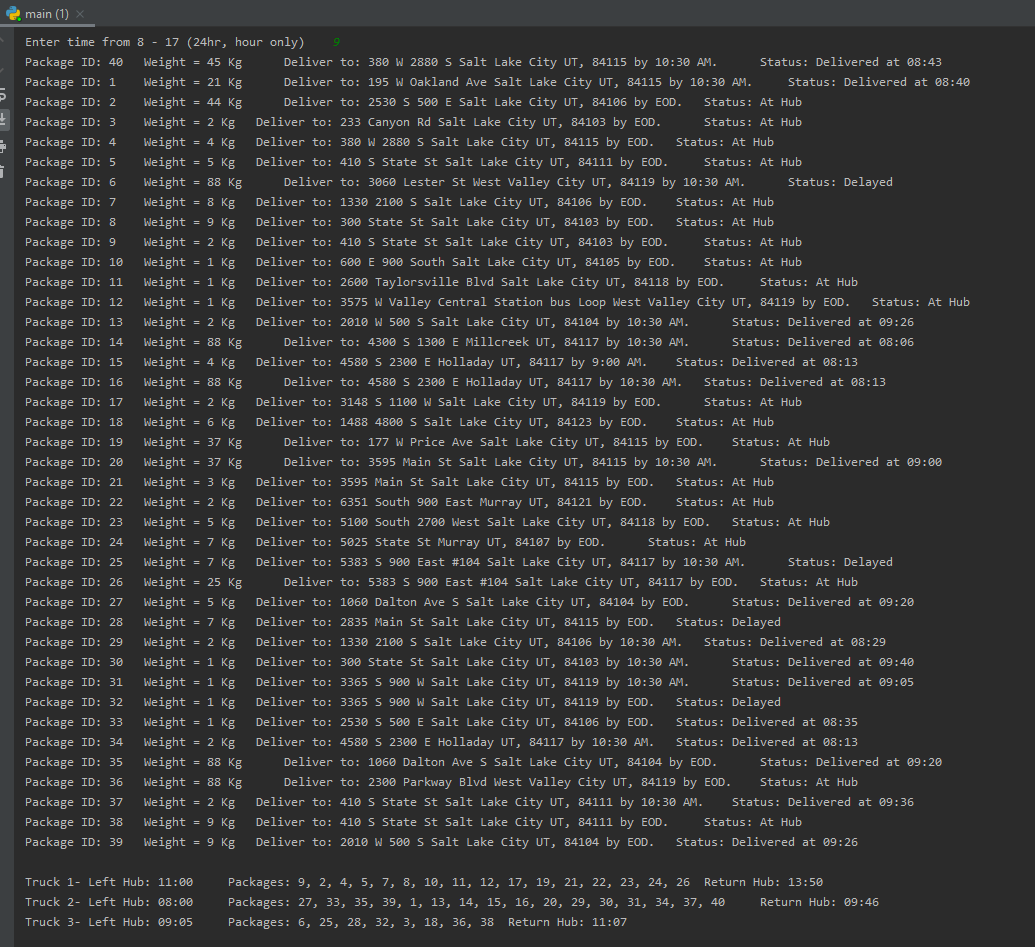
The program first loads data from the package.txt file and creates package objects which are stored in a list and inserted into a hash table. The locations are then loaded from the locations.txt file and stored in a dictionary which relates each address to its neighbors by distance. The heuristic algorithm uses the package list to sort the packages according to priority and then loads the trucks accordingly. The truck objects use dictionaries to store location information and a list of package IDs for comparison. The greedy algorithm gets the unordered route and sorts them using the locations dictionary which was loaded from the locations.txt file. The cost of the locations dictionary and package data will be O (1), since its only called once at start. The additional dictionaries created for truck routes and time tracking will cost O(n) depending on the number of packages and locations loaded.

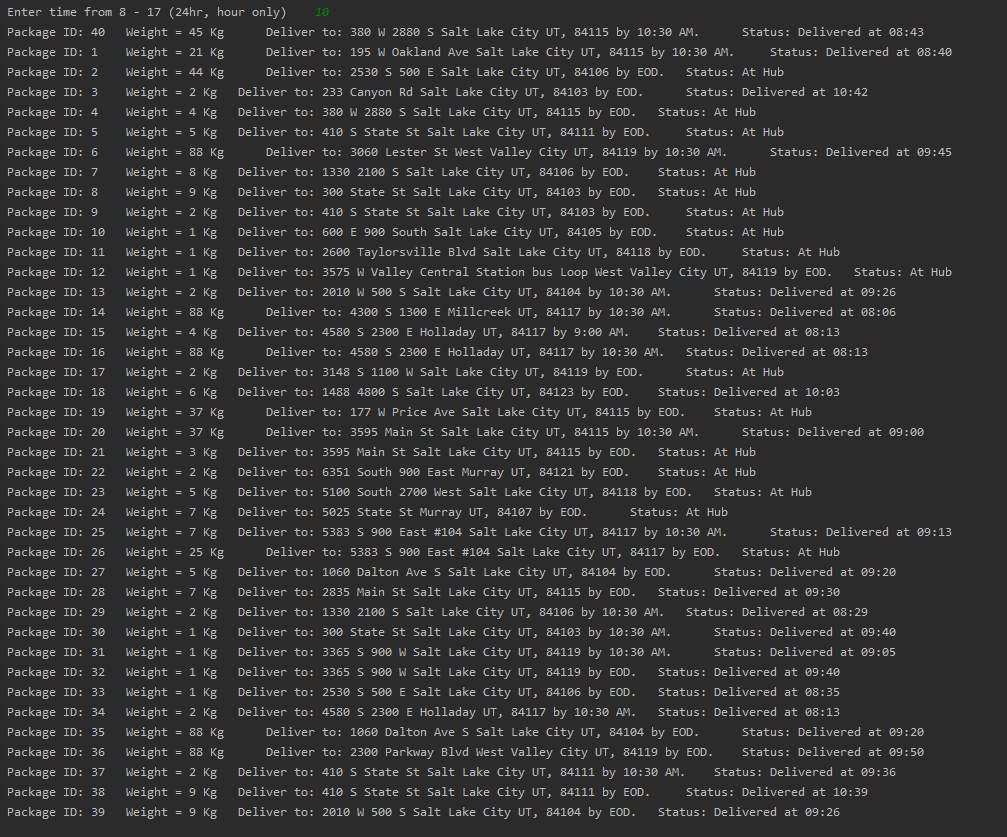
* + 1. Operation time for sorting the packages is O(n) as it loops through the package list for each condition. The operation time for finding the closest route will also be O(n). The running time for delivering the package will be O(n^2) because the code loops through the route dictionary and then loops through the delivery list to get the packages for that location. The operation time for the program is O(n^2).
    2. The program can adapt to support multiple trucks and locations. The Hash table function needs to be changed to support more packages as it uses the package ID for the key and each key is unique, therefore the table length will need to be increased to store the package data. The Heuristic algorithm could be altered if there are other special conditions for packages. The main greedy algorithm needs no alteration. If there are more locations added, the program will read the text file and store them in the dictionary accordingly. The Trucks are loaded manually so extra coding is needed to add more trucks and get more routes. Thus, the final coding depends on the trucks, locations or packages added which would take O(n).
    3. The Program is efficient and follows a simple logic to sort packages and then gets the quickest route. There are three main functions to deliver the packages. These functions can be improved to accommodate more packages and other conditions. The program contains comments for all functions.
    4. The data structure used for locations were dictionaries. Dictionaries are fast to lookup data and can be nested to support multiple data points. The search runtime for dictionaries is also O (1). The drawback was the multiple lines of code to store the data of all the locations in the location file.

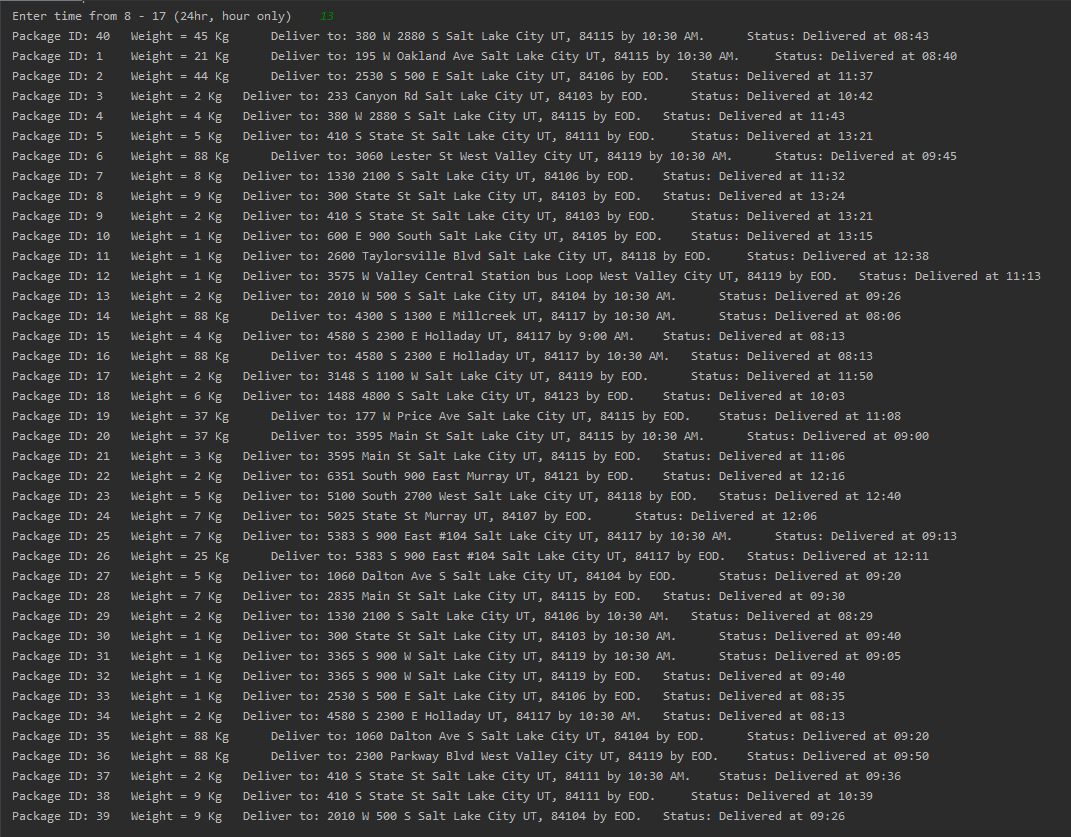
1. A Hash Table was used to store the package data. A hash table stores unordered items by mapping each item to a location in an array. The main advantage of using a hash table is that searching and inserting an item may require O (1) time. In the Hash table for the program, the key was the package ID, for all the items stored in that table. (Key % 40) hash function was used to distribute the packages into different buckets. Since there were 40 packages, each key would point to a unique bucket index thus avoiding collisions. The Search function for the hash table used the package ID as the key and returned the package object.

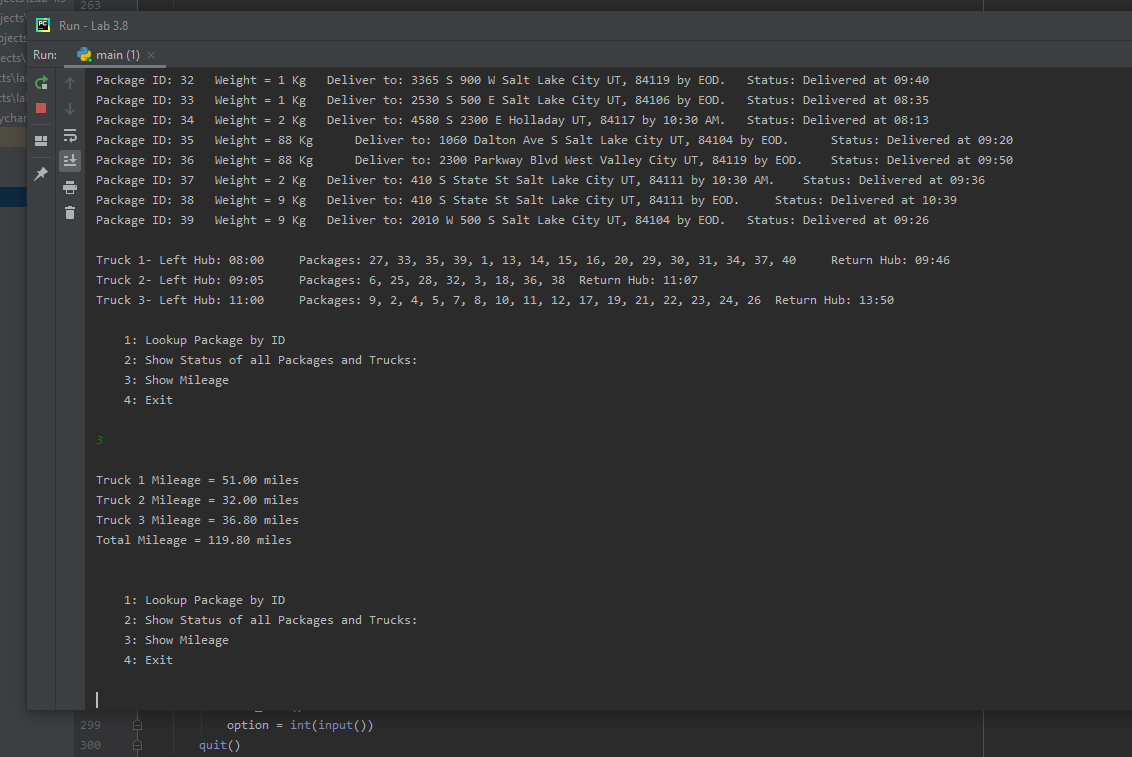
G. Status of Packages at 8 am and 9am:





Status of Packages at 10am:

Status of Packages at 1pm:

Status of Trucks and The mileage:

**Section 2:**

1. 1. The Algorithm used found the quickest route around the city and delivered all the packages before their deadlines and before the end of day. The program first used heuristics to sort the packages according to their level of priorities i.e. packages with deadlines, specific instructions, delayed and then EOD. The main advantage of the heuristic algorithm is that it improves execution speed by quickly selecting an approximate or near optimal solution. This requires less computational resources and improves speed. The trucks leave the hub accordingly and the greedy algorithm then finds the nearest location from the hub and then recursively calls that function for each nearest location in order to get the quickest route. An advantage of the greedy algorithm is that it is easier to implement. It finds the optimal choice at that time, so for every nearest location, it finds the next closest location and so forth. Another advantage of the greedy algorithm is that its easier to analyze run time when compared to other techniques (like divide and conquer). The operation time for the greedy algorithm was O(n). In general, a greedy algorithm tends to be more simple than other algorithms and efficient at finding the most optimal solution at that point in time.

2. The Packages were delivered on time and the total mileage at the end of the day was 119.8 miles.

3. Dijkstras algorithm and Breadth- first search could be used to find the route.

BFS uses vertices and queues to find the route. The starting vertex is pushed to the queue, while the queue is not empty, the algorithm removes the first vertex from queue and visits that vertex, pushing the vertex’s’ adjacent vertices and repeats.

Dijkstras algorithm finds the shortest path from a starting vertex to each vertex in the graph. For every vertex, the algorithm determines the distance and the previous vertex. The vertex’s distance is the shortest path from the starting vertex.

J. If I did this project again, I would use Dijkstras algorithm to find the route and I would add more conditions in the sorting packages function to increase efficiency such as organizing packages by zip code and city.

K. 1. The data structures I used were Dictionaries and Hash Table.

a. The hash table is an array of pointers to packages which are stored in buckets. The hash function: (key % N) where N is the number of packages, calculated the index for the bucket and returned the package object. Since the package IDs were unique, they were used for the key. Hash tables are generally more efficient than search trees and have a time complexity of O (1) for both searching and inserting.

b. The overhead for both hash table and dictionaries were minimum because there were no pointers or other variable to look up data. The hash table was created manually with a simple hash function. The size of the table was known beforehand and there was no need for any collision resolution, thus, minimizing memory bandwidth. The lookup and search functions were O (1) and the space complexity of the hash table depends on the number of packages.

c. The hash table’s size could be increased since the number of packages is known beforehand. The table size will need to be added manually for any increase in packages which will cost O (1). The trucks objects are created manually and are independent of the packages. The locations are also independent and stored in another dictionary. If the number of locations is increased, it will be appended to the text file and loaded automatically by the program in the locations dictionary.

2. Doubly-linked lists and sets could be used for location and package delivery. Doubly linked list Implements a list ADT where every node has data, and pointers to other nodes. Sets hold primitive data values and distinguish elements based on that elements key value like hash tables.

**Sources:**

Roman Lysecky, F. V. (2018). C950: Data Structures and Algorithms II. zyBooks.