**DSA2 WGUPS Program Planning Part 2**

Part D:

9:00 a.m.

A screenshot of a computer program

Description automatically generated

Figure 1

10:20 a.m. A screenshot of a computer program

Description automatically generated

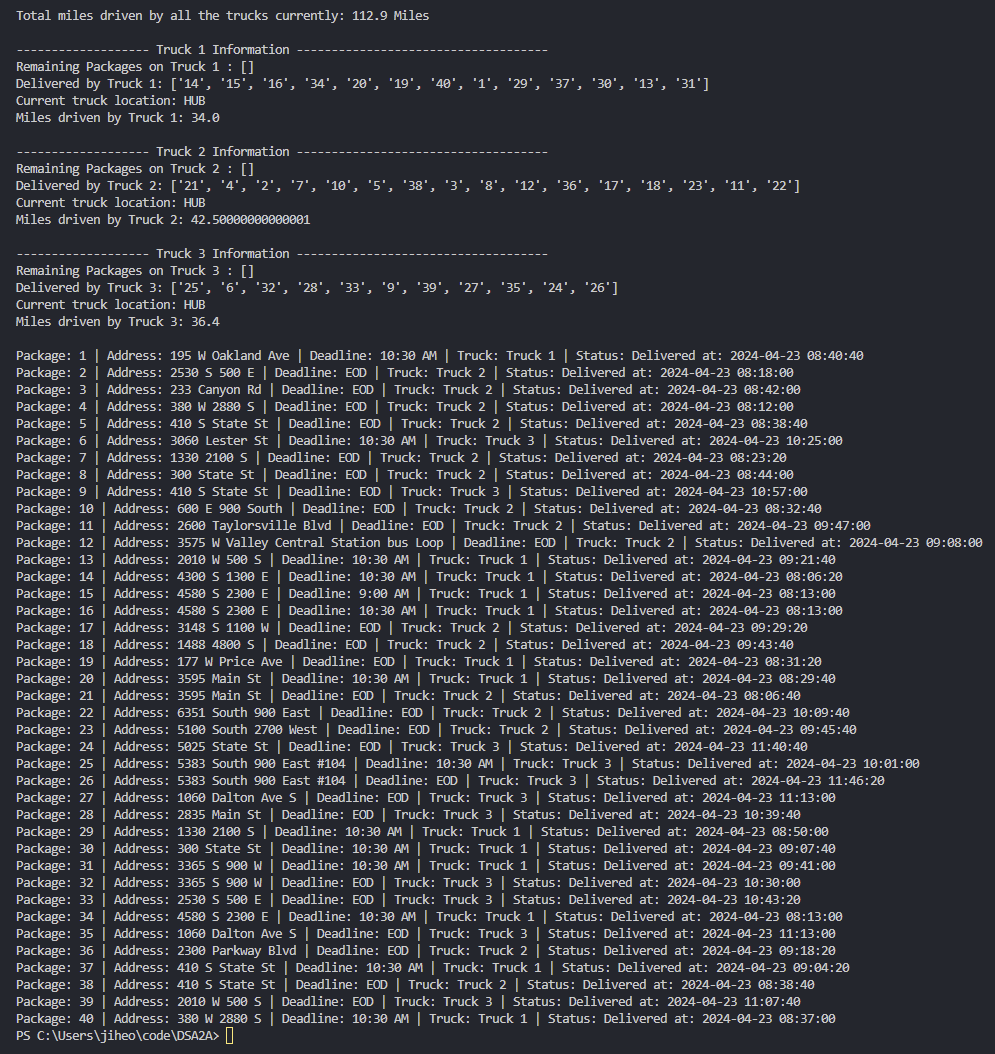
Figure 2

1:00 p.m. A screenshot of a computer program

Description automatically generated

Figure 3

Part E:



(Figure 4)

Part F: Justify the package delivery algorithm used in the solution as written in the original program by doing the following.

1. Describe **two or more** strengths of the algorithm used in the solution.
   1. One strength of the algorithm used here is that it utilizes a hash map (dictionary) to efficiently manage package data. Each delivery requires the package data to be updated. However, using a hash map as the main data structure for maintaining the information allows those updates to be made in O (1).
   2. Another strength of the algorithm used is it’s capability to accommodate more addresses and it’s ability to use the distance chart. After the initial build of the Distance matrix, which takes O(N^2) time, the access of the distance is O(1). The address is retrieved from the package O(1) and used to retrieve it’s corresponding key from the address dictionary O(1). This key is used to retrieve the data from the distance matrix accessed by [i][j] which also take O(1) time. Hence, no matter how large the amount of addresses get, the calculation of finding the nearest route will always be have a runtime complexity of O(N) to loop through the list of packages.
2. Verify that the algorithm used in the solution meets all requirements in the scenario.
   1. The algorithm starts to deliver the packages on the truck. If packages are in the priority list, those packages are delivered first using the nearest neighbor algorithm. Once all the priority packages are delivered, the algorithm continues to deliver the packages with no deadline until all packages are left and returns to the hub. Upon return of one Truck/Driver, Truck 3 departs delivering in the same process as the other Trucks.
   2. Each truck can carry 16 packages maximum.
      1. Figure 4 shows that no more than 16 packages are delivered by each truck.
      2. Truck 1 = 13
         1. Contains priority packages.
      3. Truck 2 = 16
         1. Contains EOD packages and other special cases.
      4. Truck 3 = 11
         1. Contains late arrival package and wrong address package.
   3. 3 trucks and 2 drivers
      1. Referring to figure 4
         1. Truck 1 and Truck 2 start to deliver at 8 a.m.
            1. See. Package 14 and 21 delivered around 8:06
            2. Truck 1’s last package 31 is delivered at 9:41
         2. Truck 3’s first package (25) is delivered at 10:01 since the driver of Truck 1 returns and leaves with Truck 3. Since departure time is past 9:41, all the remaining packages were already loaded onto the truck
         3. Truck 3 finishes with package 26 at 11:46
   4. Drivers can leave the hub no earlier than 8:00 a.m.
      1. Truck 1 and Truck 2 leave at 8 a.m.
   5. Special Cases
      1. Can only be on truck 2.
         1. Packages 3, 18, 36, 38 could only be on truck 2 and were delivered on truck 2.
      2. Delayed flight.
         1. Packages 6, 25, 28, 32 were delayed on flight so they were loaded into Truck 3 which left the hub after 9:41
      3. Wrong Address
         1. Package 9 had a wrong address but was updated to the correct address at 10:20 and subsequently delivered at 10:57
      4. Must be delivered with
         1. Packages 13, 14, 15, 16, 19, 20 had to be delivered together, they were all delivered on Truck 1
   6. Deadlines
      1. 9:00 a.m.
         1. Package #15 had to be delivered before 9 a.m. and was delivered at 8:13
      2. 10:30 a.m.
         1. #1
            1. 8:40 am
         2. #6
            1. 10:25 am
         3. #13
            1. 9:21 am
         4. #14
            1. 8:06 am
         5. #16
            1. 8:13 am
         6. #20
            1. 8:29 am
         7. #25
            1. 10:01 am
         8. #29
            1. 8:50 am
         9. #30
            1. 9:07 am
         10. #31
             1. 9:41 am
         11. #34
             1. 8:13 am
         12. #37
             1. 9:04 am
         13. #40
             1. 8:37 am
3. Identify two other named algorithms that are different from the algorithm implemented in the solution and would meet all requirements in the scenario.
   1. Brute Force: Another algorithm that could’ve worked to efficiently deliver all the packages would be to create a route for every possible route with the given packages on the truck. This algorithm would theoretically have an O(N^N) since the number of possible roots would be N! for a given number of packages. However, once the computation is done, we will have the route with the least distance travelled. This is different from the heuristic nearest neighbor algorithm used since it brute forces a fastest route by checking every option without considering distances between nodes.
   2. Ant Colony optimization: Can be used to train for paths that typically have shorter distances. After a certain number of iterations, the solution will continue to improve as the “ants” continue to use paths that were successful (successful indicating that they were fast tours). This is also different from the nearest neighbor algorithm since an optimization process takes place before the trip.
      1. Source: <https://www.geeksforgeeks.org/introduction-to-ant-colony-optimization/>

Part G: Describe what you would do differently, other than the two algorithms identified in part F3, if you did this project again, including details of the modifications that would be made.

* One modification I would make to the algorithm if I were to do it again would be to factor in address zip code as well. Zip codes by design group address into a proximity. By creating a map of zip codes and average distance from a zip code area to a different zip code area, trucks could be loaded more efficiently to deliver items that will be near each other. It would also mean that the nearest neighbor algorithm would work better since the packages were loaded for them to be close to each other.
* During the nearest neighbor algorithm, an additional check could be made if the nearest neighbor will be an address within the same zip code. It should check for packages remaining in the current zip code before moving onto the next package.

Part H: Verify that the data structure used in the solution meets all requirements in the scenario.

* The hash map used for the package met all the requirements since the package was able to be retrieved from the dictionary using the package ID. The package was a class that contained all the information about the package such as its address, deadline, city, zip code, weight, status. The packages were spread out relatively evenly using a hash function.
* Identify two other data structures that could meet the same requirements in the scenario.
  + Since the package ID was in ascending order, a List could’ve been used to store all instances of the package. However, this would mean that the package\_id would always have to be an integer and no other type of key could be used if the algorithm was built using a List.
  + A set would also work for this case because no 2 packages are the same and would get added to the set. The set could be partitioned into subsets to hold packages with similar/same characteristics such as address or deadlines. However, this might lead to uneven distribution of data within the set that would increase the access time for packages with many related packages. The find method for a set would have a runtime of O(N) at the worst case since it would have to look through every item.