NHP2 TASK 1: WGUPS ROUTING PROGRAM

DATA STRUCTURES AND ALGORITHMS II – C950

John Funk

Student ID: 010010682

A: ALGORITHM SELECTION

The self-adjusting algorithm I used is similar to the nearest neighbor algorithm. I pass the full array of remaining packages to a function that determines which package is the closest to the current location. The truck then goes to that location, and the package is deleted from the array of remaining packages. The function then accepts the smaller array and repeats the calculations. The function accepts an array, removes an element, and adjusts itself to run on the new, shortened array. After all packages have been delivered, the truck returns to the hub.

B1: LOGIC COMMENTS

- Call truck_route function with truck_packages array, leave_time_hour, and leave_time_minute
    - Set current_truck_location to 0
    - Set distance_traveled to 0
    - For i in truck_packages array length
        - Set package_on_way to truck_package[i]
        - Set package_on_way.status to “On the way”
        - Insert package back into hash table
    - For i in truck_packages array length
        - Call find_shortest_distance with current_location and the truck_packages array
            - Set shortest distance to 50
            - Set current_position_distance_array to the distance_array for the current_location
            - For i in truck_packages_array_length
                - Set location to the package’s address
                - Call assign_location_code to set the location_code
                - If the distance is less than shortest_distance, set shortest_distance, next_package_index, and closest_location_code
            - Set next_location with next_package_index, closest_location_code, and shortest_distance
            - Return next_location
        - Add the shortest_distance value to distance_traveled
        - Calculate time_spent_minutes by dividing distance_traveled by 18
        - Calculate total_minutes by adding time_spent_minutes to leave_time_minute
        - If total_minutes is 60 or greater, adjust hours
        - Set current_time with hour and minute values
        - Set package_to_delete
        - Call remove_complete_packages to remove the delivered package from the array
        - Set current_truck_location
    - Call return_to_hub to set distance_traveled
    - Return distance_traveled

B2: DEVELOPMENT ENVIRONMENT

- IDE: PyCharm 2022.2.3 (Community Edition)
- Operating System: Windows 10 Home
- Processor: Intel(R) Core(TM) i5-4690K CPU @ 3.50GHz 3.50 GHz
- RAM: 32.0 GB

B3: SPACE-TIME AND BIG-O

The total Big-O runtime for the program is O(N^3). Code comments contain the Big-O runtime for each method. The largest contributor to Big-O runtime is in the find_shortest_distance method. This method has a Big-O runtime of O(N^2). This runtime comes from the way the method interacts with the hash table (O(N)) inside a loop (O(N^2)). The truck_route method then calls the find_shortest_distance method inside a loop. The looped find_shortest_distance method is what gives the program a Big-O complexity of O(N^3).

B4: ADAPTABILITY

One scalable element of this project is the self-adjusting algorithm. The algorithm could scale to any array length without a change to code.

A second scalable element of this project is the self-adjusting data structure used for the package table. The hash table could scale to any number of packages without a change to code.

One element of this project that is not scalable is the method of assigning packages to trucks. The package assignment is done manually, so a code change would be required to make that part of the program scalable.

B5: SOFTWARE EFFICIENCY AND MAINTAINABILITY

The total program Big-O time complexity is O(N^3). This means the program runs in polynomial time. The way the program is compartmentalized gives it strong maintainability. Future developers would easily be able to figure out which function is doing which action for the program based on the comments and function names.

B6: SELF-ADJUSTING DATA STRUCTURES

I used a chaining hash table as a self-adjusting data structure. One strength of hash tables is speed when creating and deleting data. A second strength of hash tables is that they will never run out of space. One weakness of hash tables is inefficiency dealing with collisions. Because a hash table will almost always have collisions, this can become a more significant weakness.

D: DATA STRUCTURE

The data structure I used is a chaining hash table.

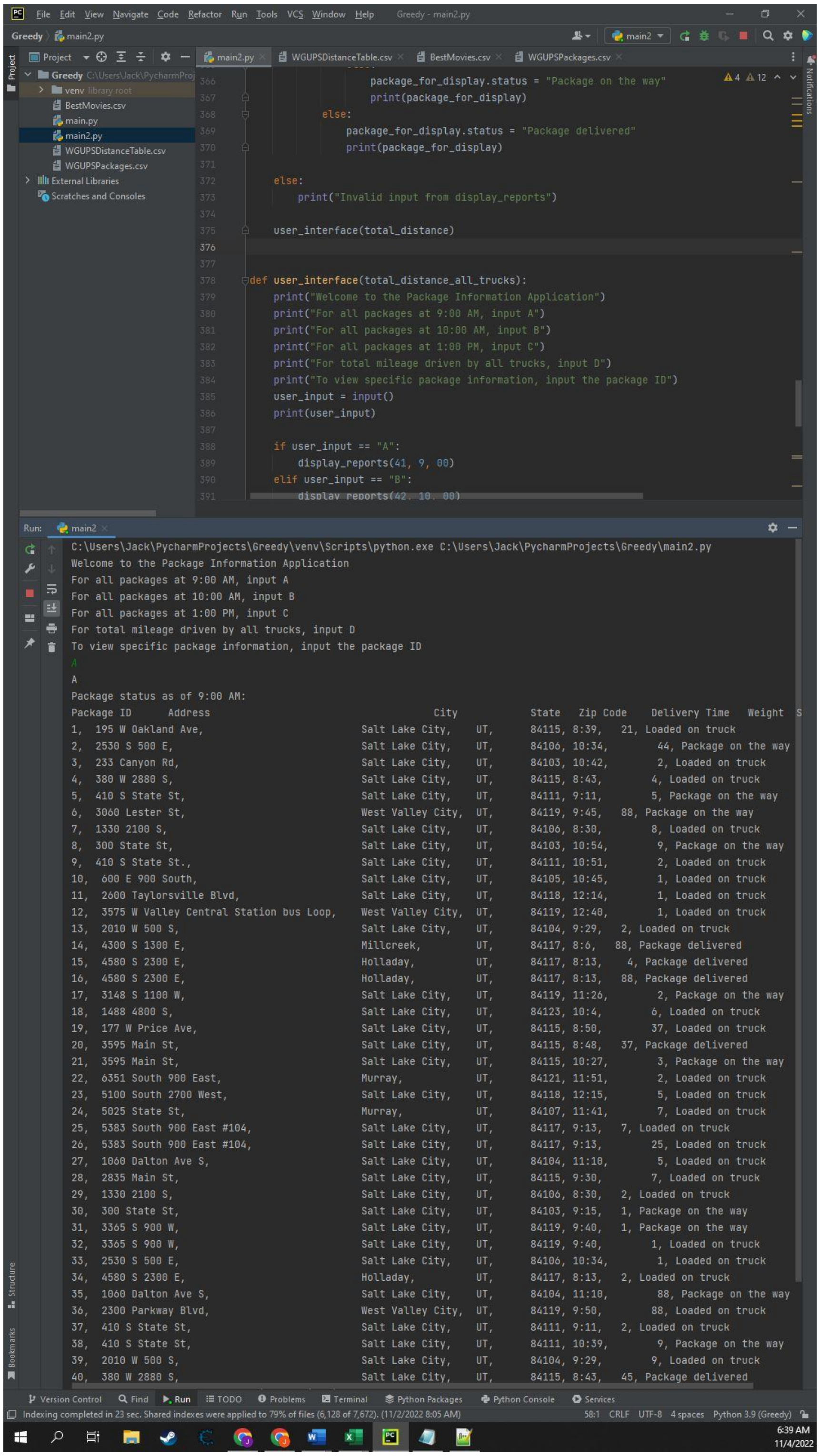
D1: EXPLANATION OF DATA STRUCTURE

The hash table stores information using the insert_package function. It creates a bucket by performing the hash and modulo functions on the insert_key variable. It then adds the bucket to the bucket list. It then appends the key and the item to the bucket list. The has table retrieves information using the search_package function. It creates a bucket by performing the hash and modulo functions on the search_key variable. If the search_key_value variable is in the bucket list, it compares the first index of the search_key_value to the search_key variable. If the two values are equal, it returns the second index of the search_key_value variable.

G: INTERFACE

I have also included the screenshots in the submission zip

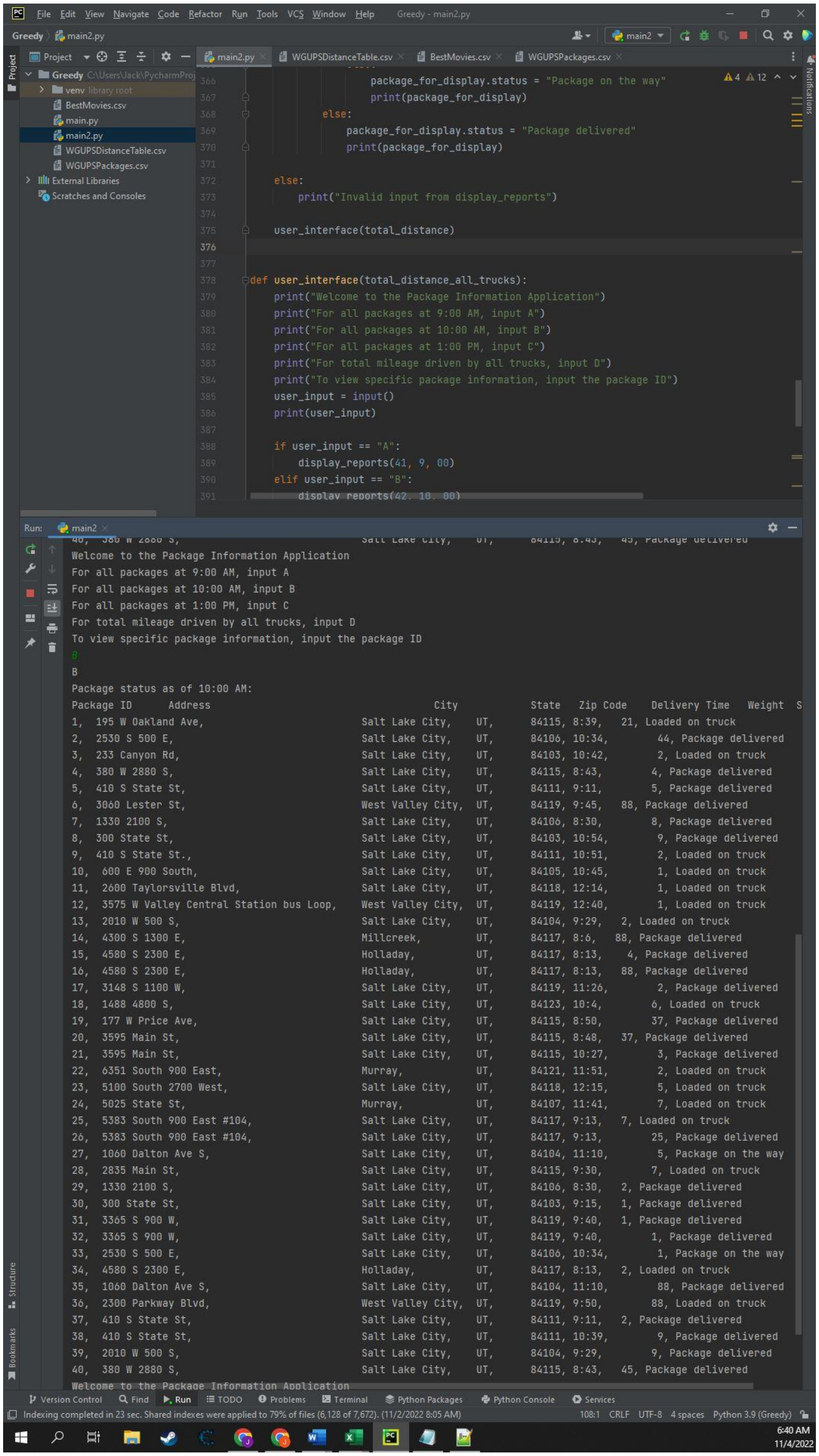
- G1 – Screenshot of the report at 9:00AM (0900 Report Screenshot.JPG)



```
                    package_for_display.status = "Package on the way"
                    print(package_for_display)
            else:
                package_for_display.status = "Package delivered"
                print(package_for_display)

    else:
        print("Invalid input from display_reports")

    user_interface(total_distance)


def user_interface(total_distance_all_trucks):
    print("Welcome to the Package Information Application")
    print("For all packages at 9:00 AM, input A")
    print("For all packages at 10:00 AM, input B")
    print("For all packages at 1:00 PM, input C")
    print("For total mileage driven by all trucks, input D")
    print("To view specific package information, input the package ID")
    user_input = input()
    print(user_input)

    if user_input == "A":
        display_reports(41, 9, 00)
    elif user_input == "B":
        display_reports(42, 10, 00)
```

```
C:\Users\Jack\PycharmProjects\Greedy\venv\Scripts\python.exe C:\Users\Jack\PycharmProjects\Greedy\main2.py
Welcome to the Package Information Application
For all packages at 9:00 AM, input A
For all packages at 10:00 AM, input B
For all packages at 1:00 PM, input C
For total mileage driven by all trucks, input D
To view specific package information, input the package ID
A
A
Package status as of 9:00 AM:
Package ID      Address                                     City            State   Zip Code    Delivery Time   Weight  S
1,  195 W Oakland Ave,                          Salt Lake City,    UT,      84115, 8:39,   21, Loaded on truck
2,  2530 S 500 E,                               Salt Lake City,    UT,      84106, 10:34,        44, Package on the way
3,  233 Canyon Rd,                              Salt Lake City,    UT,      84103, 10:42,        2, Loaded on truck
4,  380 W 2880 S,                               Salt Lake City,    UT,      84115, 8:43,        4, Loaded on truck
5,  410 S State St,                             Salt Lake City,    UT,      84111, 9:11,        5, Package on the way
6,  3060 Lester St,                             West Valley City,  UT,      84119, 9:45,   88, Package on the way
7,  1330 2100 S,                                Salt Lake City,    UT,      84106, 8:30,        8, Loaded on truck
8,  300 State St,                               Salt Lake City,    UT,      84103, 10:54,        9, Package on the way
9,  410 S State St.,                            Salt Lake City,    UT,      84111, 10:51,        2, Loaded on truck
10,  600 E 900 South,                           Salt Lake City,    UT,      84105, 10:45,        1, Loaded on truck
11,  2600 Taylorsville Blvd,                    Salt Lake City,    UT,      84118, 12:14,        1, Loaded on truck
12,  3575 W Valley Central Station bus Loop,    West Valley City,  UT,      84119, 12:40,        1, Loaded on truck
13,  2010 W 500 S,                              Salt Lake City,    UT,      84104, 9:29,   2, Loaded on truck
14,  4300 S 1300 E,                             Millcreek,         UT,      84117, 8:6,   88, Package delivered
15,  4580 S 2300 E,                             Holladay,          UT,      84117, 8:13,    4, Package delivered
16,  4580 S 2300 E,                             Holladay,          UT,      84117, 8:13,   88, Package delivered
17,  3148 S 1100 W,                             Salt Lake City,    UT,      84119, 11:26,        2, Package on the way
18,  1488 4800 S,                               Salt Lake City,    UT,      84123, 10:4,        6, Loaded on truck
19,  177 W Price Ave,                           Salt Lake City,    UT,      84115, 8:50,        37, Loaded on truck
20,  3595 Main St,                              Salt Lake City,    UT,      84115, 8:48,   37, Package delivered
21,  3595 Main St,                              Salt Lake City,    UT,      84115, 10:27,        3, Package on the way
22,  6351 South 900 East,                       Murray,            UT,      84121, 11:51,        2, Loaded on truck
23,  5100 South 2700 West,                      Salt Lake City,    UT,      84118, 12:15,        5, Loaded on truck
24,  5025 State St,                             Murray,            UT,      84107, 11:41,        7, Loaded on truck
25,  5383 South 900 East #104,                  Salt Lake City,    UT,      84117, 9:13,   7, Loaded on truck
26,  5383 South 900 East #104,                  Salt Lake City,    UT,      84117, 9:13,        25, Loaded on truck
27,  1060 Dalton Ave S,                         Salt Lake City,    UT,      84104, 11:10,        5, Loaded on truck
28,  2835 Main St,                              Salt Lake City,    UT,      84115, 9:30,        7, Loaded on truck
29,  1330 2100 S,                               Salt Lake City,    UT,      84106, 8:30,   2, Loaded on truck
30,  300 State St,                              Salt Lake City,    UT,      84103, 9:15,   1, Package on the way
31,  3365 S 900 W,                              Salt Lake City,    UT,      84119, 9:40,   1, Package on the way
32,  3365 S 900 W,                              Salt Lake City,    UT,      84119, 9:40,        1, Loaded on truck
33,  2530 S 500 E,                              Salt Lake City,    UT,      84106, 10:34,        1, Loaded on truck
34,  4580 S 2300 E,                             Holladay,          UT,      84117, 8:13,   2, Loaded on truck
35,  1060 Dalton Ave S,                         Salt Lake City,    UT,      84104, 11:10,        88, Package on the way
36,  2300 Parkway Blvd,                         West Valley City,  UT,      84119, 9:50,        88, Loaded on truck
37,  410 S State St,                            Salt Lake City,    UT,      84111, 9:11,   2, Loaded on truck
38,  410 S State St,                            Salt Lake City,    UT,      84111, 10:39,        9, Package on the way
39,  2010 W 500 S,                              Salt Lake City,    UT,      84104, 9:29,        9, Loaded on truck
40,  380 W 2880 S,                              Salt Lake City,    UT,      84115, 8:43,   45, Package delivered
```

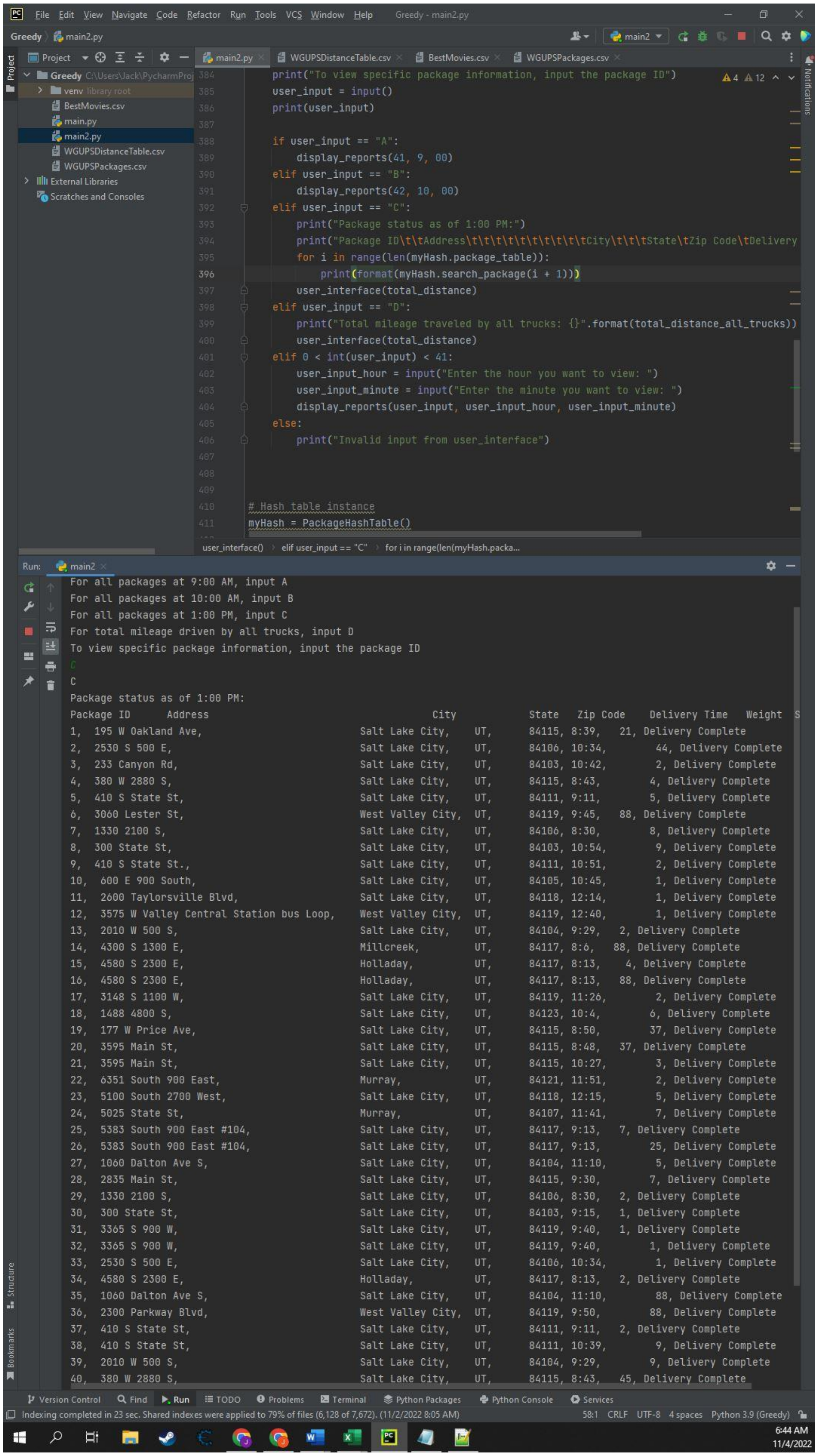
- G2 – Screenshot of the report at 10:00 AM (1000 Report Screenshot.JPG)



```python
                    package_for_display.status = "Package on the way"
                    print(package_for_display)
            else:
                package_for_display.status = "Package delivered"
                print(package_for_display)

    else:
        print("Invalid input from display_reports")

    user_interface(total_distance)


def user_interface(total_distance_all_trucks):
    print("Welcome to the Package Information Application")
    print("For all packages at 9:00 AM, input A")
    print("For all packages at 10:00 AM, input B")
    print("For all packages at 1:00 PM, input C")
    print("For total mileage driven by all trucks, input D")
    print("To view specific package information, input the package ID")
    user_input = input()
    print(user_input)

    if user_input == "A":
        display_reports(41, 9, 00)
    elif user_input == "B":
        display_reports(42, 10, 00)
```

```
40,  380 W 2880 S,                              Salt Lake City,    UT,      84115, 8:43,   45, Package delivered
Welcome to the Package Information Application
For all packages at 9:00 AM, input A
For all packages at 10:00 AM, input B
For all packages at 1:00 PM, input C
For total mileage driven by all trucks, input D
To view specific package information, input the package ID
B
B
Package status as of 10:00 AM:
Package ID      Address                                     City           State   Zip Code    Delivery Time   Weight  S
1,  195 W Oakland Ave,                          Salt Lake City,    UT,      84115, 8:39,   21, Loaded on truck
2,  2530 S 500 E,                               Salt Lake City,    UT,      84106, 10:34,        44, Package delivered
3,  233 Canyon Rd,                              Salt Lake City,    UT,      84103, 10:42,        2, Loaded on truck
4,  380 W 2880 S,                               Salt Lake City,    UT,      84115, 8:43,        4, Package delivered
5,  410 S State St,                             Salt Lake City,    UT,      84111, 9:11,        5, Package delivered
6,  3060 Lester St,                             West Valley City,  UT,      84119, 9:45,   88, Package delivered
7,  1330 2100 S,                                Salt Lake City,    UT,      84106, 8:30,        8, Package delivered
8,  300 State St,                               Salt Lake City,    UT,      84103, 10:54,        9, Package delivered
9,  410 S State St.,                            Salt Lake City,    UT,      84111, 10:51,        2, Loaded on truck
10,  600 E 900 South,                           Salt Lake City,    UT,      84105, 10:45,        1, Loaded on truck
11,  2600 Taylorsville Blvd,                    Salt Lake City,    UT,      84118, 12:14,        1, Loaded on truck
12,  3575 W Valley Central Station bus Loop,    West Valley City,  UT,      84119, 12:40,        1, Loaded on truck
13,  2010 W 500 S,                              Salt Lake City,    UT,      84104, 9:29,   2, Loaded on truck
14,  4300 S 1300 E,                             Millcreek,         UT,      84117, 8:6,   88, Package delivered
15,  4580 S 2300 E,                             Holladay,          UT,      84117, 8:13,    4, Package delivered
16,  4580 S 2300 E,                             Holladay,          UT,      84117, 8:13,   88, Package delivered
17,  3148 S 1100 W,                             Salt Lake City,    UT,      84119, 11:26,        2, Package delivered
18,  1488 4800 S,                               Salt Lake City,    UT,      84123, 10:4,        6, Loaded on truck
19,  177 W Price Ave,                           Salt Lake City,    UT,      84115, 8:50,        37, Package delivered
20,  3595 Main St,                              Salt Lake City,    UT,      84115, 8:48,   37, Package delivered
21,  3595 Main St,                              Salt Lake City,    UT,      84115, 10:27,        3, Package delivered
22,  6351 South 900 East,                       Murray,            UT,      84121, 11:51,        2, Loaded on truck
23,  5100 South 2700 West,                      Salt Lake City,    UT,      84118, 12:15,        5, Loaded on truck
24,  5025 State St,                             Murray,            UT,      84107, 11:41,        7, Loaded on truck
25,  5383 South 900 East #104,                  Salt Lake City,    UT,      84117, 9:13,   7, Loaded on truck
26,  5383 South 900 East #104,                  Salt Lake City,    UT,      84117, 9:13,        25, Package delivered
27,  1060 Dalton Ave S,                         Salt Lake City,    UT,      84104, 11:10,        5, Package on the way
28,  2835 Main St,                              Salt Lake City,    UT,      84115, 9:30,        7, Loaded on truck
29,  1330 2100 S,                               Salt Lake City,    UT,      84106, 8:30,   2, Package delivered
30,  300 State St,                              Salt Lake City,    UT,      84103, 9:15,   1, Package delivered
31,  3365 S 900 W,                              Salt Lake City,    UT,      84119, 9:40,   1, Package delivered
32,  3365 S 900 W,                              Salt Lake City,    UT,      84119, 9:40,        1, Package delivered
33,  2530 S 500 E,                              Salt Lake City,    UT,      84106, 10:34,        1, Package on the way
34,  4580 S 2300 E,                             Holladay,          UT,      84117, 8:13,   2, Loaded on truck
35,  1060 Dalton Ave S,                         Salt Lake City,    UT,      84104, 11:10,        88, Package delivered
36,  2300 Parkway Blvd,                         West Valley City,  UT,      84119, 9:50,        88, Loaded on truck
37,  410 S State St,                            Salt Lake City,    UT,      84111, 9:11,   2, Package delivered
38,  410 S State St,                            Salt Lake City,    UT,      84111, 10:39,        9, Package delivered
39,  2010 W 500 S,                              Salt Lake City,    UT,      84104, 9:29,        9, Package delivered
40,  380 W 2880 S,                              Salt Lake City,    UT,      84115, 8:43,   45, Package delivered
Welcome to the Package Information Application
```

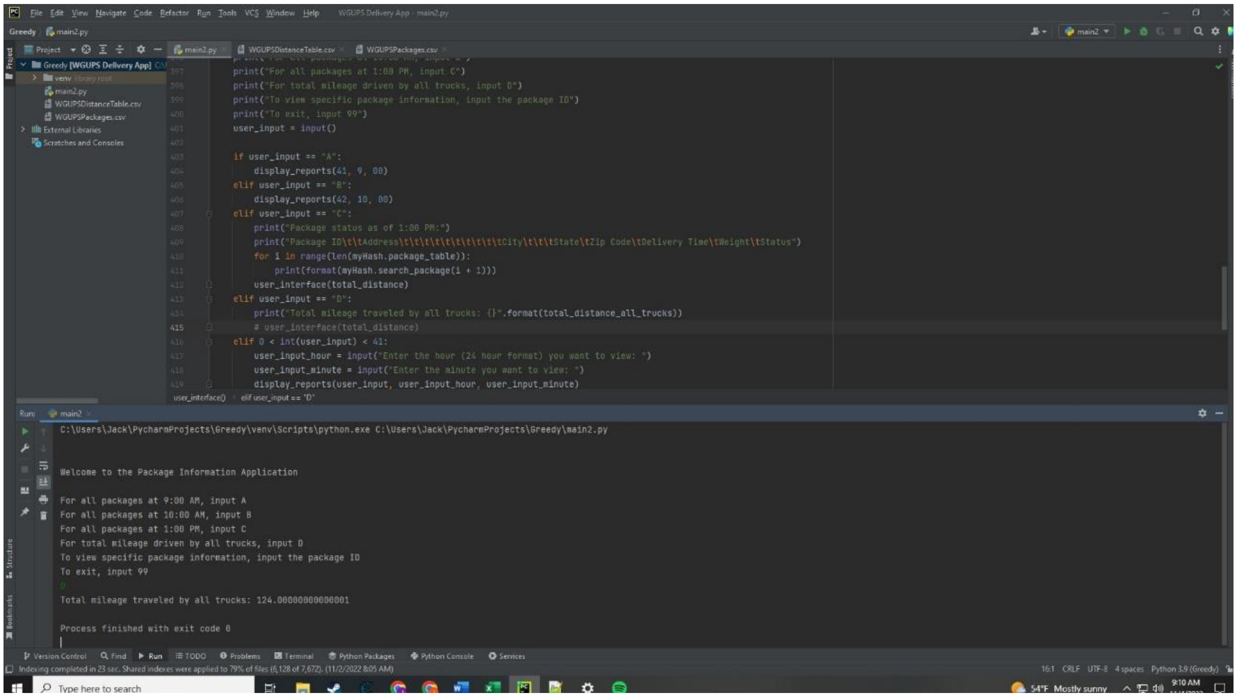
- G3 – Screenshot of the report for 1:00 PM (1300 Report Screenshot.JPG)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Greedy - main2.py
Greedy
main2.py
main2
Project
Greedy C:\Users\Jack\PycharmProj
venv library root
BestMovies.csv
main.py
main2.py
WGUPSDistanceTable.csv
WGUPSPackages.csv
External Libraries
Scratches and Consoles
4 12
Notifications
user_interface() › elif user_input == "C" › for i in range(len(myHash.packa...
Run: main2
Structure
Bookmarks
Version Control
Find
Run
TODO
Problems
Terminal
Python Packages
Python Console
Services
Indexing completed in 23 sec. Shared indexes were applied to 79% of files (6,128 of 7,672). (11/2/2022 8:05 AM)
58:1 CRLF UTF-8 4 spaces Python 3.9 (Greedy)
6:44 AM
11/4/2022

```python
    print("To view specific package information, input the package ID")
    user_input = input()
    print(user_input)

    if user_input == "A":
        display_reports(41, 9, 00)
    elif user_input == "B":
        display_reports(42, 10, 00)
    elif user_input == "C":
        print("Package status as of 1:00 PM:")
        print("Package ID\t\tAddress\t\t\t\t\t\t\t\t\t\tCity\t\t\tState\tZip Code\tDelivery
        for i in range(len(myHash.package_table)):
            print(format(myHash.search_package(i + 1)))
        user_interface(total_distance)
    elif user_input == "D":
        print("Total mileage traveled by all trucks: {}".format(total_distance_all_trucks))
        user_interface(total_distance)
    elif 0 < int(user_input) < 41:
        user_input_hour = input("Enter the hour you want to view: ")
        user_input_minute = input("Enter the minute you want to view: ")
        display_reports(user_input, user_input_hour, user_input_minute)
    else:
        print("Invalid input from user_interface")



# Hash table instance
myHash = PackageHashTable()
```

```
For all packages at 9:00 AM, input A
For all packages at 10:00 AM, input B
For all packages at 1:00 PM, input C
For total mileage driven by all trucks, input D
To view specific package information, input the package ID
C
C
Package status as of 1:00 PM:
Package ID      Address                                     City            State   Zip Code    Delivery Time   Weight  S
1,  195 W Oakland Ave,                          Salt Lake City,    UT,      84115, 8:39,   21, Delivery Complete
2,  2530 S 500 E,                               Salt Lake City,    UT,      84106, 10:34,        44, Delivery Complete
3,  233 Canyon Rd,                              Salt Lake City,    UT,      84103, 10:42,        2, Delivery Complete
4,  380 W 2880 S,                               Salt Lake City,    UT,      84115, 8:43,        4, Delivery Complete
5,  410 S State St,                             Salt Lake City,    UT,      84111, 9:11,        5, Delivery Complete
6,  3060 Lester St,                             West Valley City,  UT,      84119, 9:45,   88, Delivery Complete
7,  1330 2100 S,                                Salt Lake City,    UT,      84106, 8:30,        8, Delivery Complete
8,  300 State St,                               Salt Lake City,    UT,      84103, 10:54,        9, Delivery Complete
9,  410 S State St.,                            Salt Lake City,    UT,      84111, 10:51,        2, Delivery Complete
10,  600 E 900 South,                           Salt Lake City,    UT,      84105, 10:45,        1, Delivery Complete
11,  2600 Taylorsville Blvd,                    Salt Lake City,    UT,      84118, 12:14,        1, Delivery Complete
12,  3575 W Valley Central Station bus Loop,    West Valley City,  UT,      84119, 12:40,        1, Delivery Complete
13,  2010 W 500 S,                              Salt Lake City,    UT,      84104, 9:29,   2, Delivery Complete
14,  4300 S 1300 E,                             Millcreek,         UT,      84117, 8:6,   88, Delivery Complete
15,  4580 S 2300 E,                             Holladay,          UT,      84117, 8:13,    4, Delivery Complete
16,  4580 S 2300 E,                             Holladay,          UT,      84117, 8:13,   88, Delivery Complete
17,  3148 S 1100 W,                             Salt Lake City,    UT,      84119, 11:26,        2, Delivery Complete
18,  1488 4800 S,                               Salt Lake City,    UT,      84123, 10:4,        6, Delivery Complete
19,  177 W Price Ave,                           Salt Lake City,    UT,      84115, 8:50,        37, Delivery Complete
20,  3595 Main St,                              Salt Lake City,    UT,      84115, 8:48,   37, Delivery Complete
21,  3595 Main St,                              Salt Lake City,    UT,      84115, 10:27,        3, Delivery Complete
22,  6351 South 900 East,                       Murray,            UT,      84121, 11:51,        2, Delivery Complete
23,  5100 South 2700 West,                      Salt Lake City,    UT,      84118, 12:15,        5, Delivery Complete
24,  5025 State St,                             Murray,            UT,      84107, 11:41,        7, Delivery Complete
25,  5383 South 900 East #104,                  Salt Lake City,    UT,      84117, 9:13,   7, Delivery Complete
26,  5383 South 900 East #104,                  Salt Lake City,    UT,      84117, 9:13,        25, Delivery Complete
27,  1060 Dalton Ave S,                         Salt Lake City,    UT,      84104, 11:10,        5, Delivery Complete
28,  2835 Main St,                              Salt Lake City,    UT,      84115, 9:30,        7, Delivery Complete
29,  1330 2100 S,                               Salt Lake City,    UT,      84106, 8:30,   2, Delivery Complete
30,  300 State St,                              Salt Lake City,    UT,      84103, 9:15,   1, Delivery Complete
31,  3365 S 900 W,                              Salt Lake City,    UT,      84119, 9:40,   1, Delivery Complete
32,  3365 S 900 W,                              Salt Lake City,    UT,      84119, 9:40,        1, Delivery Complete
33,  2530 S 500 E,                              Salt Lake City,    UT,      84106, 10:34,        1, Delivery Complete
34,  4580 S 2300 E,                             Holladay,          UT,      84117, 8:13,   2, Delivery Complete
35,  1060 Dalton Ave S,                         Salt Lake City,    UT,      84104, 11:10,        88, Delivery Complete
36,  2300 Parkway Blvd,                         West Valley City,  UT,      84119, 9:50,        88, Delivery Complete
37,  410 S State St,                            Salt Lake City,    UT,      84111, 9:11,   2, Delivery Complete
38,  410 S State St,                            Salt Lake City,    UT,      84111, 10:39,        9, Delivery Complete
39,  2010 W 500 S,                              Salt Lake City,    UT,      84104, 9:29,        9, Delivery Complete
40,  380 W 2880 S,                              Salt Lake City,    UT,      84115, 8:43,   45, Delivery Complete
```

H: Total Mileage Screenshot (Total Distance Screenshot.JPG)



```
        print("For all packages at 1:00 PM, input C")
        print("For total mileage driven by all trucks, input D")
        print("To view specific package information, input the package ID")
        print("To exit, input 99")
        user_input = input()

        if user_input == "A":
            display_reports(41, 9, 00)
        elif user_input == "B":
            display_reports(42, 10, 00)
        elif user_input == "C":
            print("Package status as of 1:00 PM:")
            print("Package ID\t\tAddress\t\t\t\t\t\t\t\t\t\tCity\t\t\tState\tZip Code\tDelivery Time\tWeight\tStatus")
            for i in range(len(myHash.package_table)):
                print(format(myHash.search_package(i + 1)))
            user_interface(total_distance)
        elif user_input == "D":
            print("Total mileage traveled by all trucks: {}".format(total_distance_all_trucks))
            # user_interface(total_distance)
        elif 0 < int(user_input) < 41:
            user_input_hour = input("Enter the hour (24 hour format) you want to view: ")
            user_input_minute = input("Enter the minute you want to view: ")
            display_reports(user_input, user_input_hour, user_input_minute)
```

```
C:\Users\Jack\PycharmProjects\Greedy\venv\Scripts\python.exe C:\Users\Jack\PycharmProjects\Greedy\main2.py


Welcome to the Package Information Application

For all packages at 9:00 AM, input A
For all packages at 10:00 AM, input B
For all packages at 1:00 PM, input C
For total mileage driven by all trucks, input D
To view specific package information, input the package ID
To exit, input 99
D
Total mileage traveled by all trucks: 124.00000000000001

Process finished with exit code 0
```

Indexing completed in 23 sec. Shared indexes were applied to 79% of files (6,128 of 7,672). (11/2/2022 8:05 AM)

I1: STRENGTHS OF THE CHOSEN ALGORITHM

One strength of the chosen algorithm is simplicity. It is very easy to understand, and that makes it easy to use. A second strength of this algorithm is flexibility. The algorithm can accept an input array of any size and still function.

I3: OTHER POSSIBLE ALGORITHMS

Two other algorithms that could have solved this problem are the Brute Force Approach and Dynamic Programming.

I3A: ALGORITHM DIFFERENCES

The Brute Force Approach calculates all possible routes a truck could take. It then compares all possible routes to find the shortest one. The Brute Force Approach has a time complexity of O(N!) (Solonen, 2018), so it is much more time intensive than my method. Dynamic Programming is a method that guarantees finding the optimal route. The time complexity of Dynamic Programming is O(N^2 X 2^N), so it is also much more time intensive than my method. (Ataee, 2020)

J: DIFFERENT APPROACH

If I reattempted the project, I would have used a datetime style variable to control the required delivery time and the actual delivery time. I had to convert the delivery time variables to datetime date types throughout the project.

K1: VERIFICATION OF DATA STRUCTURE

The total combined miles traveled by all trucks was 124. All packages were delivered on time. All packages were delivered according to their delivery specifications. An efficient hash table with a lookup function is present. The reporting can be verified through the user interface and all information is accurate.

K1A: EFFICIENCY

The lookup function for the hash table runs at O(N). This means that the time is affected linearly by an increase in number of packages.

K1B: OVERHEAD

As the number of packages increases, the number of items in the linked list will increase at the same rate.

K1C: IMPLICATIONS

As trucks increase, the hash table is not affected. The hash table does not store information that has to do with trucks. As the number of cities increases, the hash table is not affected. The hash table only stores the city name, so adding different names would not affect lookup time or space usage.

K2: OTHER DATA STRUCTURES

One data structure that could meet the requirements in the scenario is a dictionary. A second data structure that could meet the requirements in the scenario is a stack.

K2A: DATA STRUCTURES DIFFERENCES

- A dictionary stores information as key/value pairs, whereas a hash table stores information as objects. A dictionary allows the use of prebuilt methods, whereas I had to write code to access the hash table.
- A stack can only be accessed from one end of the stack, whereas a hash table can be accessed throughout the table. A stack allows the use of prebuilt methods, whereas I had to write code to access the hash table.

L: SOURCES

- Solonen, A. (2018). *4.4.2. NP-hard problems¶*. 4.4.2. NP-hard problems - Learn Programming 1.0. Retrieved November 4, 2022, from https://progbook.org/tsp.html#:~:text=The%20brute%20force%20search%20to,is%20120%20*%206%20%3D%20720.
- Ataee, P. (2020, June 14). *How to solve traveling salesman problem — a comparative analysis*. Towards Data Science. Retrieved November 4, 2022, from https://towardsdatascience.com/how-to-solve-the-traveling-salesman-problem-a-comparative-analysis-39056a916c9f