C950 WGUPS Algorithm Overview

Justin Golay
ID #010176782
WGU Email: jgolay@wgu.edu

February 5, 2023

C950 Data Structures and Algorithms II

Introduction

The purpose of this project is to create an algorithm using Python programming language to develop an effective and efficient solution to deliver packages. There are three delivery trucks, two drivers, and 40 packages. Each delivery truck can only hold 16 packages at one time. Limitations to the delivery of these packages come in the form of delivery deadlines, special instructions for various packages, and packages that arrive late to the mailing hub and must then depart later in the day than other packages.

This program utilizes the Nearest Neighbor Algorithm to simulate package delivery. The end user can then enter a specific time and determine the delivery status of an individual package or all packages at that time.

A. Algorithm Identification

This program utilizes the Nearest Neighbor Algorithm to order package deliveries.

B1. Logic Comments

The Nearest Neighbor Algorithm is created in the following manner:

1. Create an empty list and assign all package objects to it.
2. Utilize a while loop to continue until this list is empty
3. Initiate a variable referring to the package’s next address as a number greater than any of the distances to be compared in the delivery simulation.
4. Initiate a variable referring to the next package to be compared and set it to none.
5. Utilize a for loop to compare each package’s destination with the current location and update the variables to reflect the shortest distance.
6. Place the package with the destination that is the closest to its closest location onto the truck.
7. Remove the package from the initial package list.
8. Repeat the process until the initial list of packages contains no packages.

Pseudocode:

```
function nearestNeighborSearchAlgorithm(truck):
    unsortedPackages = empty list
    FOR packages on truck:
        APPEND packages to unsortedPackages
```

```
CLEAR packages from truck


WHILE length of unsorted packages > 0:
    nextAddress = infinity
    nextPackage = None
    FOR packages in unsortedPackages:
        truckAddress = getAddress (truck address)
        eachPackageAddress = getAddress(eachPackage address)
        IF distanceBetweenCities(truckAddress, eachPackageAddress) <=
        nextAddress:
            nextAddress = distanceBetweenCities(truckAddress,
            eachPackageAddress)
            nextPackage = eachPackage
    APPEND nextPackage to truck
    REMOVE nextPackage from unsortedPackages
    UPDATE truck mileage += nextAddress
    UPDATE truck address += nextPackage address
    UPDATE truck time += nextAddress/18
    UPDATE nextPackage deliveryTime = truck time
    UPDATE nextPackage departTime = truck departTime
```

Methods used in this algorithm include getAddress() to find the integer index of a location based on its address as a string and distancesBetweenCities() takes two address indexes and locates them in a list of distances.

```
function getAddress(string address):
    FOR each row in a list of addresses:
        IF address is in row
            RETURN integer index number


function distanceBetweenCities(xValue, yValue):
    distanceBetweenCities = distance list indexed at xValue, yValue
    IF distanceBetweenCities == '':
        distanceBetweenCities = distance list indexed at yValue, xValue
    RETURN float distanceBetweenCities
```

B2. Development Environment

The program was developed utilizing PyCharm Community Edition and Python version 3.11. The operating system was Windows 11 Home version 22H2. The system uses an AMD Ryzen 5 processor and 64-bit operating system.

B3. Space-Time and Big-O

The Nearest Neighbor Algorithm of this program has a time complexity of O(N^2). The overall program has a time complexity of O(N^2). Further, the program creates lists utilized by the Nearest Neighbor Algorithm that have a space complexity of O(N). The overall space complexity of the program is O(N).

B4. Scalability and Adaptability

This program utilizes a hash table to organize packages by ID number. This results in a package look up time complexity of O(1). However, this hash table was developed with 40 buckets, which is efficient for this program but may be less efficient should the program take a much larger number of packages. The number of buckets in the hash table can be increased to improve efficiency with larger inputs.

Also, given the time complexity of this Nearest Neighbor Algorithm is O(N^2), a larger number of packages to deliver would take longer (quadratic growth) for the program compute than if it were given a smaller number of packages.

If the delivery service were to expand operations, additional servers may need to be purchased to store package information and address lists, which would grow linearly. However, delivery trucks are limited in the number of packages that they can haul due to the size of the truck. This limitation of the delivery truck means that it is unlikely the company would need to purchase faster servers to calculate delivery routes.

B5. Software Efficiency and Maintainability

This software is efficient and easy to maintain because it is based on a commonly known algorithm called the Nearest Neighbor Algorithm with a time complexity of O(N^2) ("Nearest Neighbour Algorithm"). Developers can reference pseudocode to ensure that the algorithm maintains function even if the company utilizing the software is required to switch to another programming language.

The Nearest Neighbor Algorithm is more time efficient than other approaches to similar problems such as the Brute Force approach, which would utilize a time complexity of O(N!) (Chase et al.).

B6. Self-Adjusting Data Structures

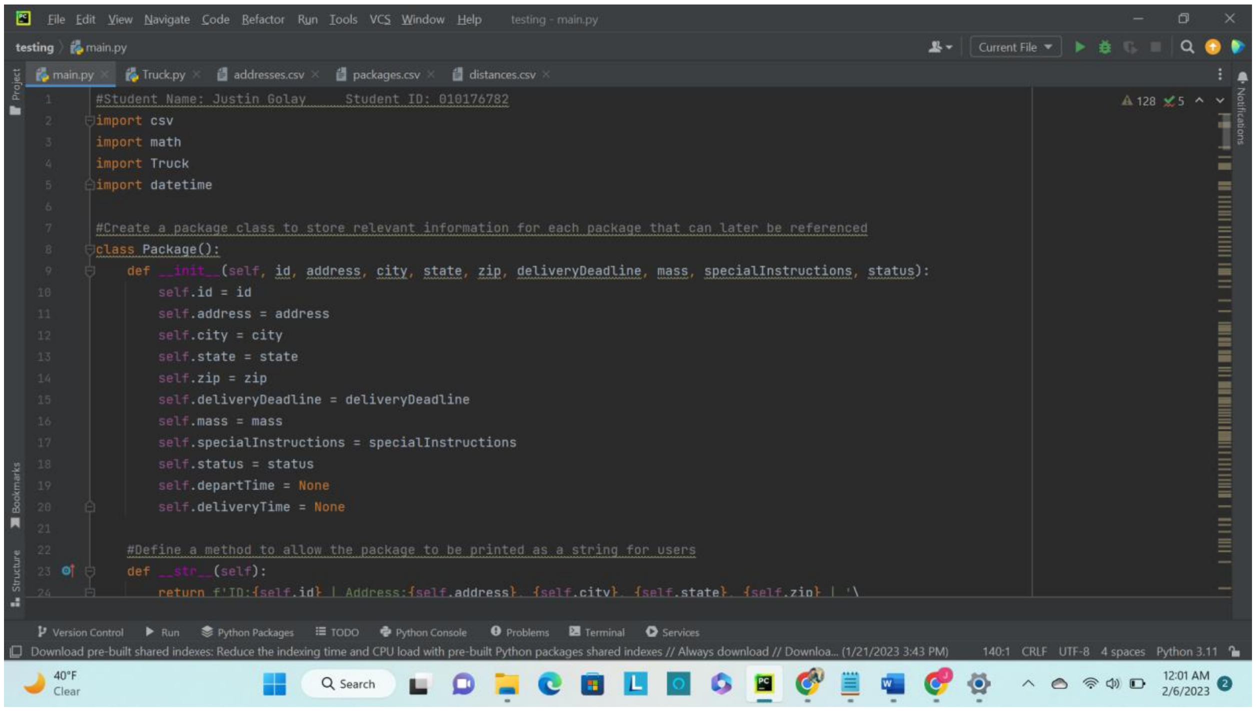
The self-adjusting data structure in this program includes the hash table. The primary benefit of utilizing a hash table for this type of problem is that it searches and updates an item in constant time. However, the primary weakness of this hash table is that its space complexity is O(N), which means that it will grow linearly as its input grows. This could result in significantly more memory being utilized should the program simulate a greater number of packages being delivered.

C. Original Code

The code for this program was completed and includes the following files to execute correctly: main.py, Truck.py, addresses.csv, packages.csv, distances.csv.

C1. Identification Information

Identification information was included on Line 1 of main.py including: Justin Golay ID #010176782.

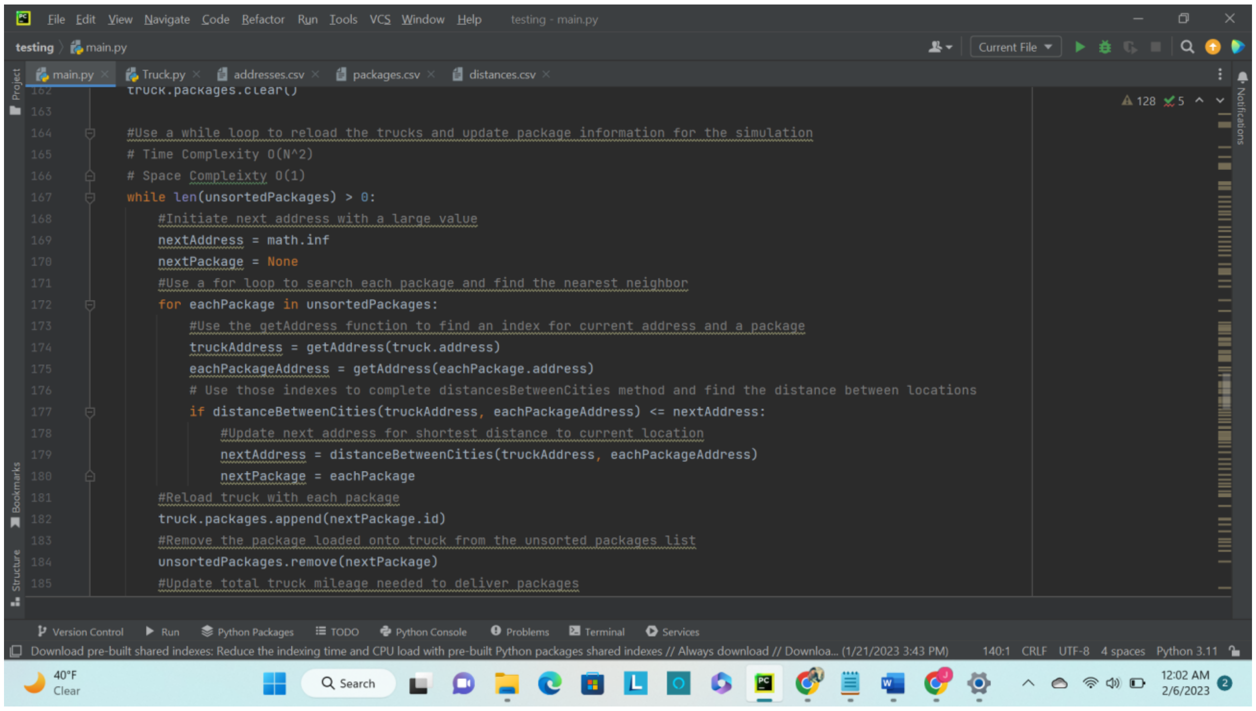


```
1   #Student Name: Justin Golay     Student ID: 010176782
2   import csv
3   import math
4   import Truck
5   import datetime
6
7   #Create a package class to store relevant information for each package that can later be referenced
8   class Package():
9       def __init__(self, id, address, city, state, zip, deliveryDeadline, mass, specialInstructions, status):
10          self.id = id
11          self.address = address
12          self.city = city
13          self.state = state
14          self.zip = zip
15          self.deliveryDeadline = deliveryDeadline
16          self.mass = mass
17          self.specialInstructions = specialInstructions
18          self.status = status
19          self.departTime = None
20          self.deliveryTime = None
21
22      #Define a method to allow the package to be printed as a string for users
23      def __str__(self):
24          return f'ID:{self.id} | Address:{self.address}, {self.city}, {self.state}, {self.zip} | '\
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Downloa... (1/21/2023 3:43 PM)

C2. Process and Flow Comments

Comments were added throughout the program to ensure that other software developers could read and understand the process and flow of this program. Below is an example:



```
truck.packages.clear()

#Use a while loop to reload the trucks and update package information for the simulation
# Time Complexity O(N^2)
# Space Compleixty O(1)
while len(unsortedPackages) > 0:
    #Initiate next address with a large value
    nextAddress = math.inf
    nextPackage = None
    #Use a for loop to search each package and find the nearest neighbor
    for eachPackage in unsortedPackages:
        #Use the getAddress function to find an index for current address and a package
        truckAddress = getAddress(truck.address)
        eachPackageAddress = getAddress(eachPackage.address)
        # Use those indexes to complete distancesBetweenCities method and find the distance between locations
        if distanceBetweenCities(truckAddress, eachPackageAddress) <= nextAddress:
            #Update next address for shortest distance to current location
            nextAddress = distanceBetweenCities(truckAddress, eachPackageAddress)
            nextPackage = eachPackage
    #Reload truck with each package
    truck.packages.append(nextPackage.id)
    #Remove the package loaded onto truck from the unsorted packages list
    unsortedPackages.remove(nextPackage)
    #Update total truck mileage needed to deliver packages
```

D. Data Structure

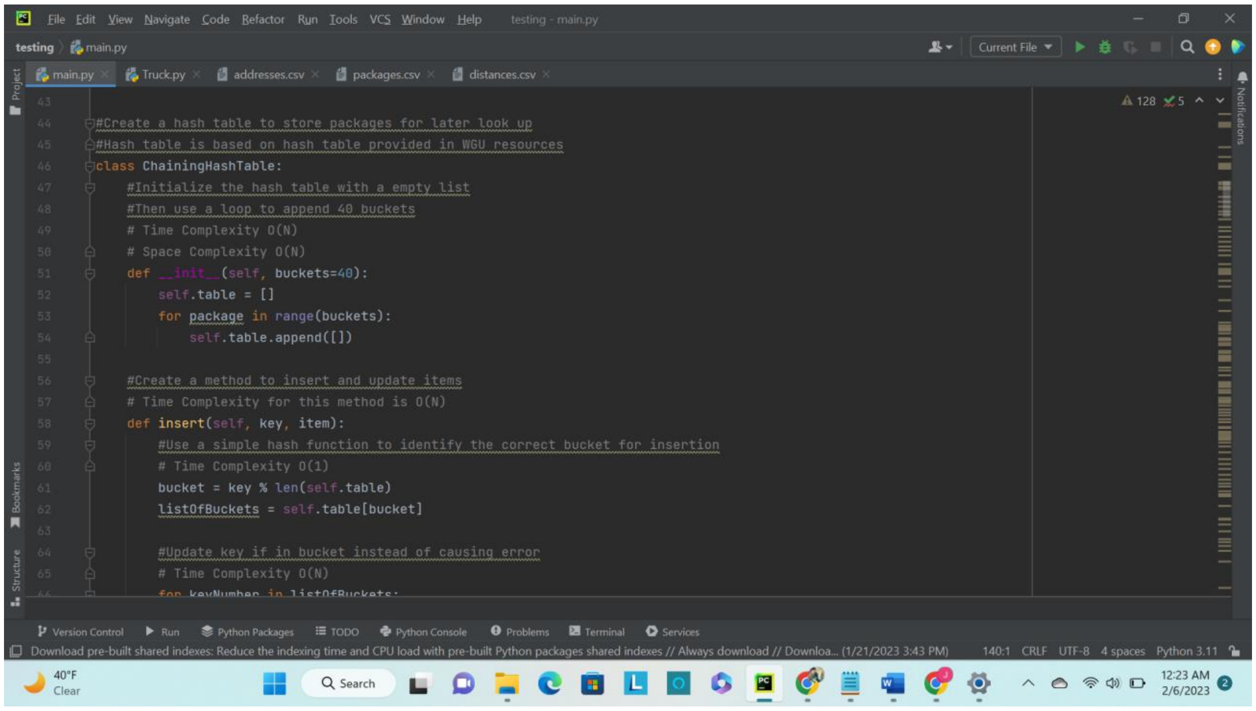
The self-adjusting data structure in this project is the hash table utilized to store package objects that store package information. They can be searched in the hash table utilizing each package's ID number as the key. Should a function call for insertion of two items with the same ID number, a collision is avoided by updating the item with the second insertion.

D1. Explanation of Data Structure

Each package delivered in this program was read from a .csv file and placed into a package object. Each package object was stored in the hash table based on its unique package ID number. The program frequently searches for packages to execute functions required to complete the Nearest Neighbor Algorithm. This hash table allows for packages to be searched in constant time.

E. Hash Table

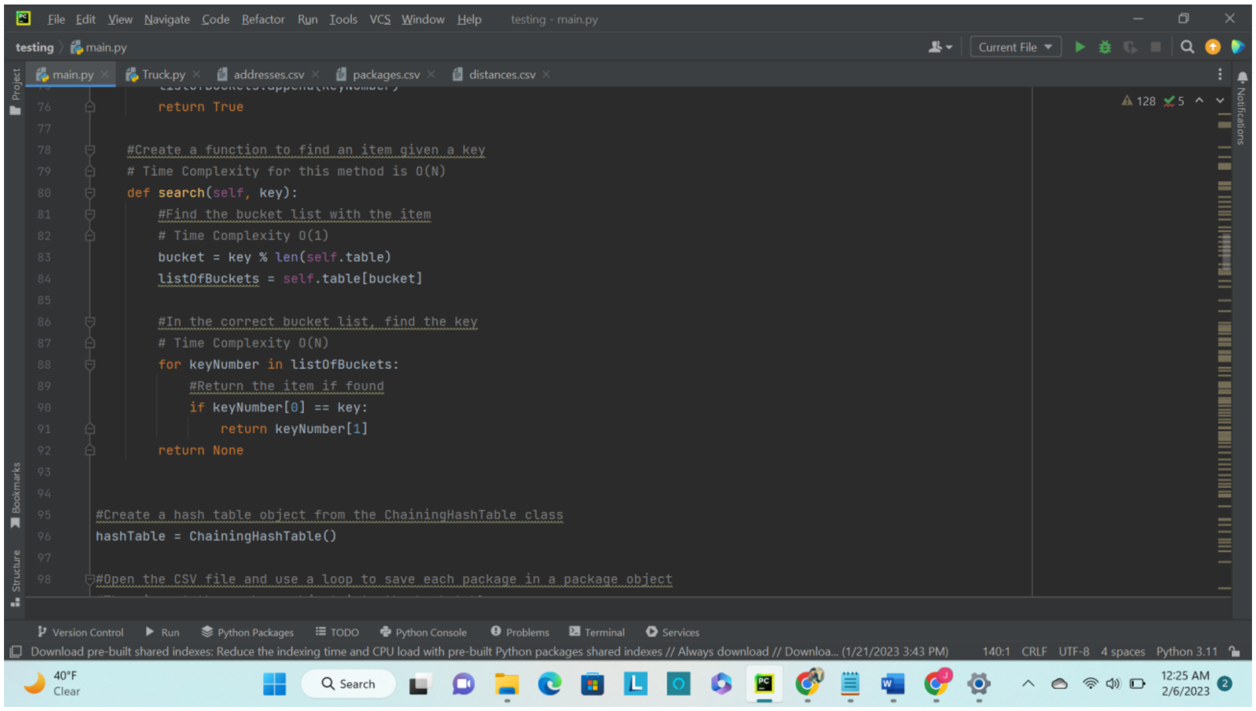
The hash table was utilized in this program. It was designed based on WGU resources (*C950 - Webinar-1 - Let's Go Hashing*).



```
#Create a hash table to store packages for later look up
#Hash table is based on hash table provided in WGU resources
class ChainingHashTable:
    #Initialize the hash table with a empty list
    #Then use a loop to append 40 buckets
    # Time Complexity O(N)
    # Space Complexity O(N)
    def __init__(self, buckets=40):
        self.table = []
        for package in range(buckets):
            self.table.append([])

    #Create a method to insert and update items
    # Time Complexity for this method is O(N)
    def insert(self, key, item):
        #Use a simple hash function to identify the correct bucket for insertion
        # Time Complexity O(1)
        bucket = key % len(self.table)
        listOfBuckets = self.table[bucket]

        #Update key if in bucket instead of causing error
        # Time Complexity O(N)
        for keyNumber in listOfBuckets:
```

F. Look-Up Function

The look up function was created in the hash table. It is called “search”. Each package can be searched in the hash table. Once that package has been found, specified information for each package can be found (i.e., ID number, address, delivery deadline, city, zip code, weight, and status.



```
            listOfBuckets.append(keyNumber)
        return True

    #Create a function to find an item given a key
    # Time Complexity for this method is O(N)
    def search(self, key):
        #Find the bucket list with the item
        # Time Complexity O(1)
        bucket = key % len(self.table)
        listOfBuckets = self.table[bucket]

        #In the correct bucket list, find the key
        # Time Complexity O(N)
        for keyNumber in listOfBuckets:
            #Return the item if found
            if keyNumber[0] == key:
                return keyNumber[1]
        return None


#Create a hash table object from the ChainingHashTable class
hashTable = ChainingHashTable()

#Open the CSV file and use a loop to save each package in a package object
```

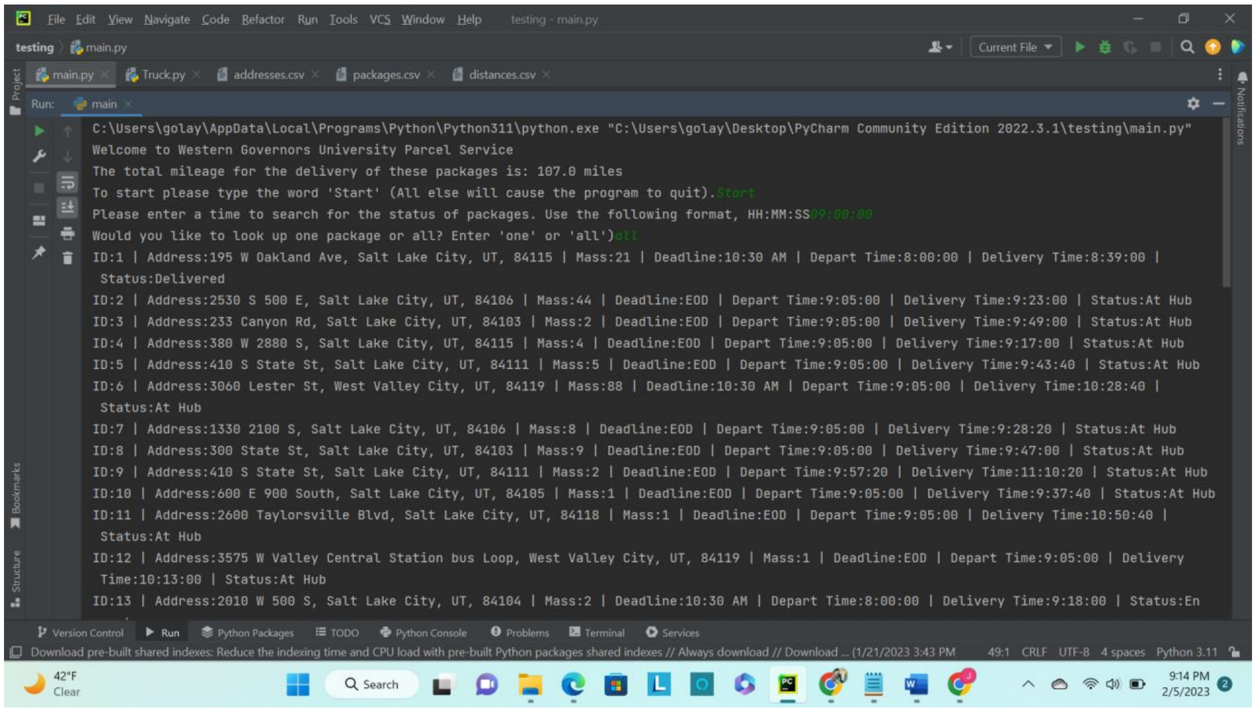
Furthermore, end users that are not familiar with coding can utilize the user interface and request to view the information for all packages at any time.

G. Interface

A user interface was developed for this program to allow users to look up any or all packages at any time. The total mileage for all trucks is displayed for users as well. Below are screen shots to demonstrate this functionality.

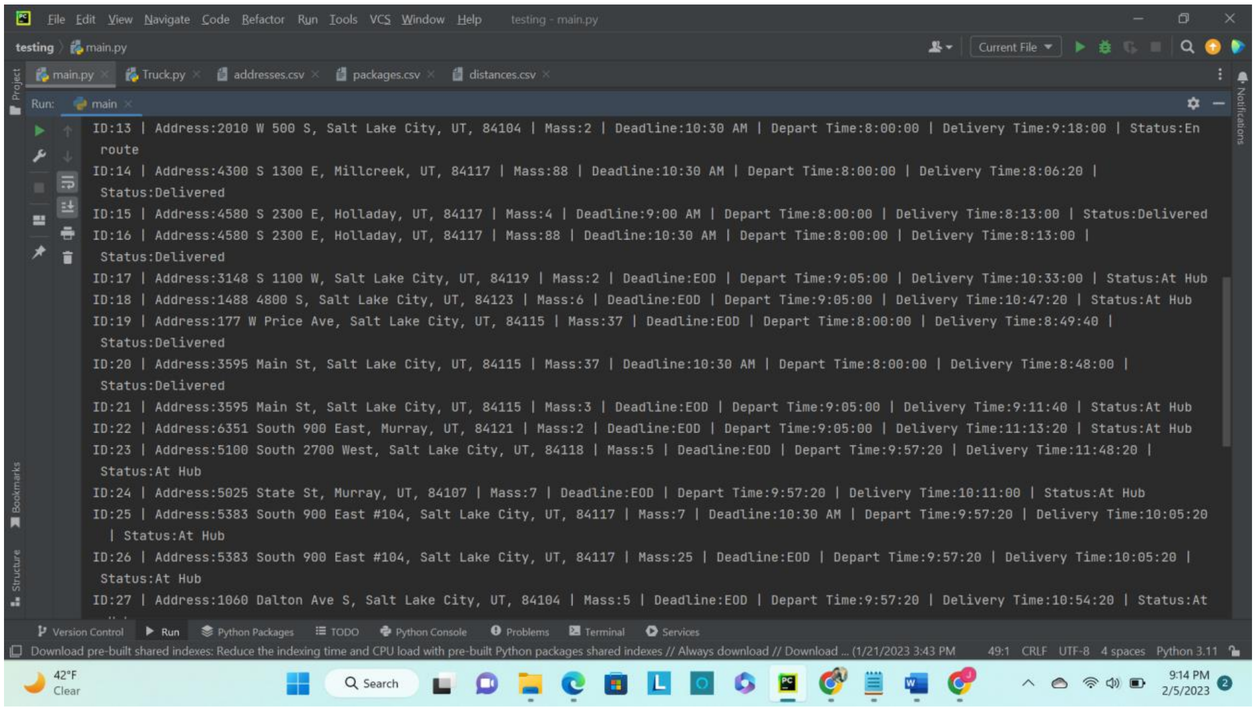
G1. First Status Check

Package Information at 9:00 a.m.

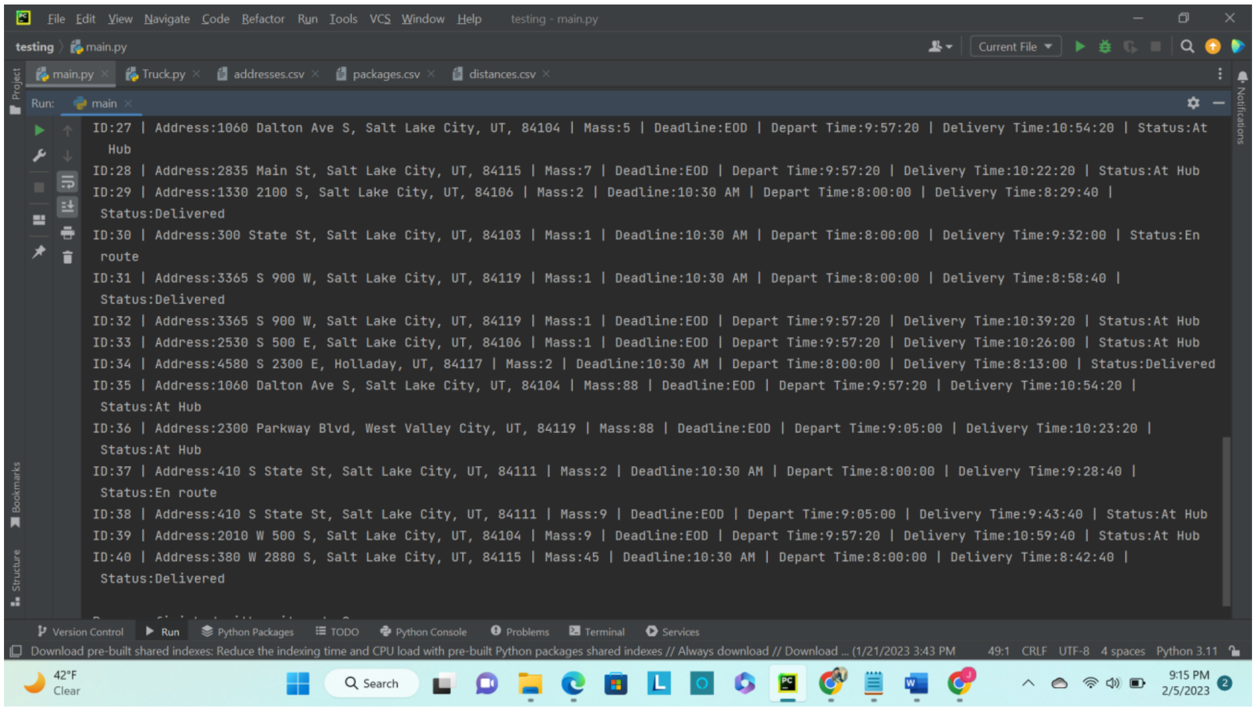


```
C:\Users\golay\AppData\Local\Programs\Python\Python311\python.exe "C:\Users\golay\Desktop\PyCharm Community Edition 2022.3.1\testing\main.py"
Welcome to Western Governors University Parcel Service
The total mileage for the delivery of these packages is: 107.0 miles
To start please type the word 'Start' (All else will cause the program to quit).Start
Please enter a time to search for the status of packages. Use the following format, HH:MM:SS09:00:00
Would you like to look up one package or all? Enter 'one' or 'all')all
ID:1 | Address:195 W Oakland Ave, Salt Lake City, UT, 84115 | Mass:21 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:39:00 |
 Status:Delivered
ID:2 | Address:2530 S 500 E, Salt Lake City, UT, 84106 | Mass:44 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:23:00 | Status:At Hub
ID:3 | Address:233 Canyon Rd, Salt Lake City, UT, 84103 | Mass:2 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:49:00 | Status:At Hub
ID:4 | Address:380 W 2880 S, Salt Lake City, UT, 84115 | Mass:4 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:17:00 | Status:At Hub
ID:5 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:5 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:43:40 | Status:At Hub
ID:6 | Address:3060 Lester St, West Valley City, UT, 84119 | Mass:88 | Deadline:10:30 AM | Depart Time:9:05:00 | Delivery Time:10:28:40 |
 Status:At Hub
ID:7 | Address:1330 2100 S, Salt Lake City, UT, 84106 | Mass:8 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:28:20 | Status:At Hub
ID:8 | Address:300 State St, Salt Lake City, UT, 84103 | Mass:9 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:47:00 | Status:At Hub
ID:9 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:2 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:11:10:20 | Status:At Hub
ID:10 | Address:600 E 900 South, Salt Lake City, UT, 84105 | Mass:1 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:37:40 | Status:At Hub
ID:11 | Address:2600 Taylorsville Blvd, Salt Lake City, UT, 84118 | Mass:1 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:50:40 |
 Status:At Hub
ID:12 | Address:3575 W Valley Central Station bus Loop, West Valley City, UT, 84119 | Mass:1 | Deadline:EOD | Depart Time:9:05:00 | Delivery
 Time:10:13:00 | Status:At Hub
ID:13 | Address:2010 W 500 S, Salt Lake City, UT, 84104 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:9:18:00 | Status:En
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
testing - main.py
main.py Truck.py addresses.csv packages.csv distances.csv
Run: main
Version Control Run Python Packages TODO Python Console Problems Terminal Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download ... (1/21/2023 3:43 PM
49:1 CRLF UTF-8 4 spaces Python 3.11
42°F Clear
9:14 PM 2/5/2023



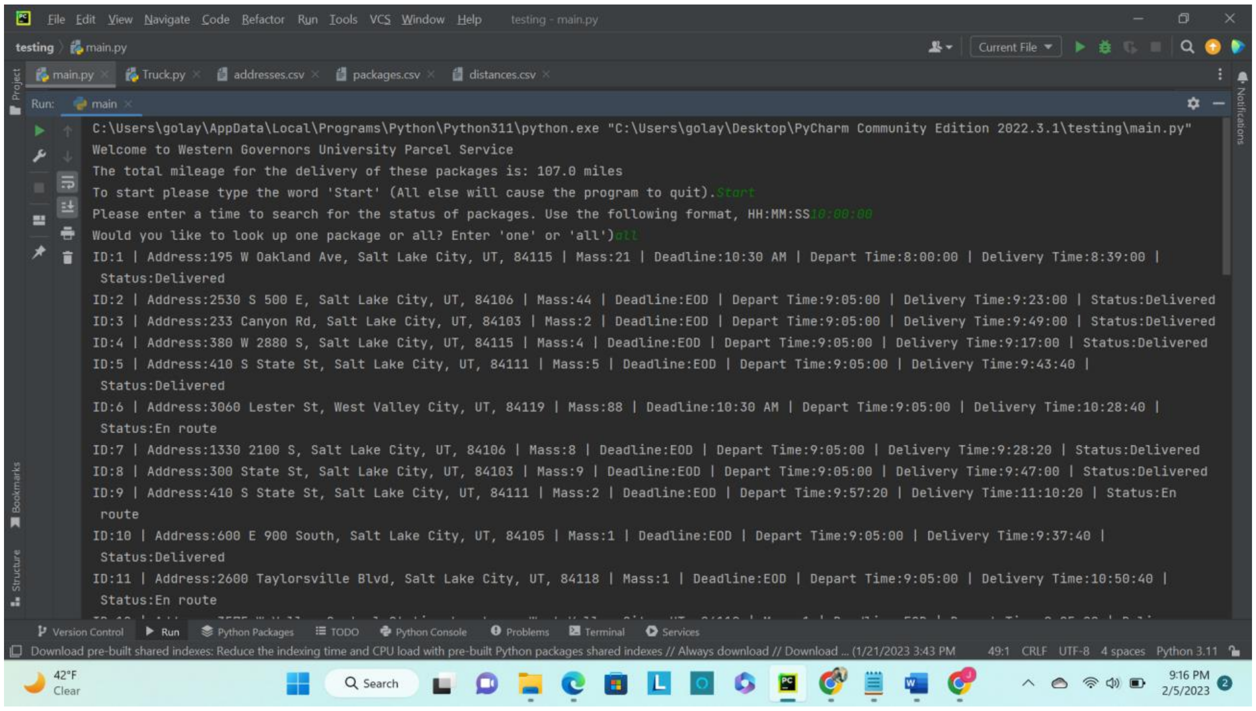
```
ID:13 | Address:2010 W 500 S, Salt Lake City, UT, 84104 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:9:18:00 | Status:En
 route
ID:14 | Address:4300 S 1300 E, Millcreek, UT, 84117 | Mass:88 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:06:20 |
 Status:Delivered
ID:15 | Address:4580 S 2300 E, Holladay, UT, 84117 | Mass:4 | Deadline:9:00 AM | Depart Time:8:00:00 | Delivery Time:8:13:00 | Status:Delivered
ID:16 | Address:4580 S 2300 E, Holladay, UT, 84117 | Mass:88 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:13:00 |
 Status:Delivered
ID:17 | Address:3148 S 1100 W, Salt Lake City, UT, 84119 | Mass:2 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:33:00 | Status:At Hub
ID:18 | Address:1488 4800 S, Salt Lake City, UT, 84123 | Mass:6 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:47:20 | Status:At Hub
ID:19 | Address:177 W Price Ave, Salt Lake City, UT, 84115 | Mass:37 | Deadline:EOD | Depart Time:8:00:00 | Delivery Time:8:49:40 |
 Status:Delivered
ID:20 | Address:3595 Main St, Salt Lake City, UT, 84115 | Mass:37 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:48:00 |
 Status:Delivered
ID:21 | Address:3595 Main St, Salt Lake City, UT, 84115 | Mass:3 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:11:40 | Status:At Hub
ID:22 | Address:6351 South 900 East, Murray, UT, 84121 | Mass:2 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:11:13:20 | Status:At Hub
ID:23 | Address:5100 South 2700 West, Salt Lake City, UT, 84118 | Mass:5 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:11:48:20 |
 Status:At Hub
ID:24 | Address:5025 State St, Murray, UT, 84107 | Mass:7 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:11:00 | Status:At Hub
ID:25 | Address:5383 South 900 East #104, Salt Lake City, UT, 84117 | Mass:7 | Deadline:10:30 AM | Depart Time:9:57:20 | Delivery Time:10:05:20
  | Status:At Hub
ID:26 | Address:5383 South 900 East #104, Salt Lake City, UT, 84117 | Mass:25 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:05:20 |
 Status:At Hub
ID:27 | Address:1060 Dalton Ave S, Salt Lake City, UT, 84104 | Mass:5 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:54:20 | Status:At
```



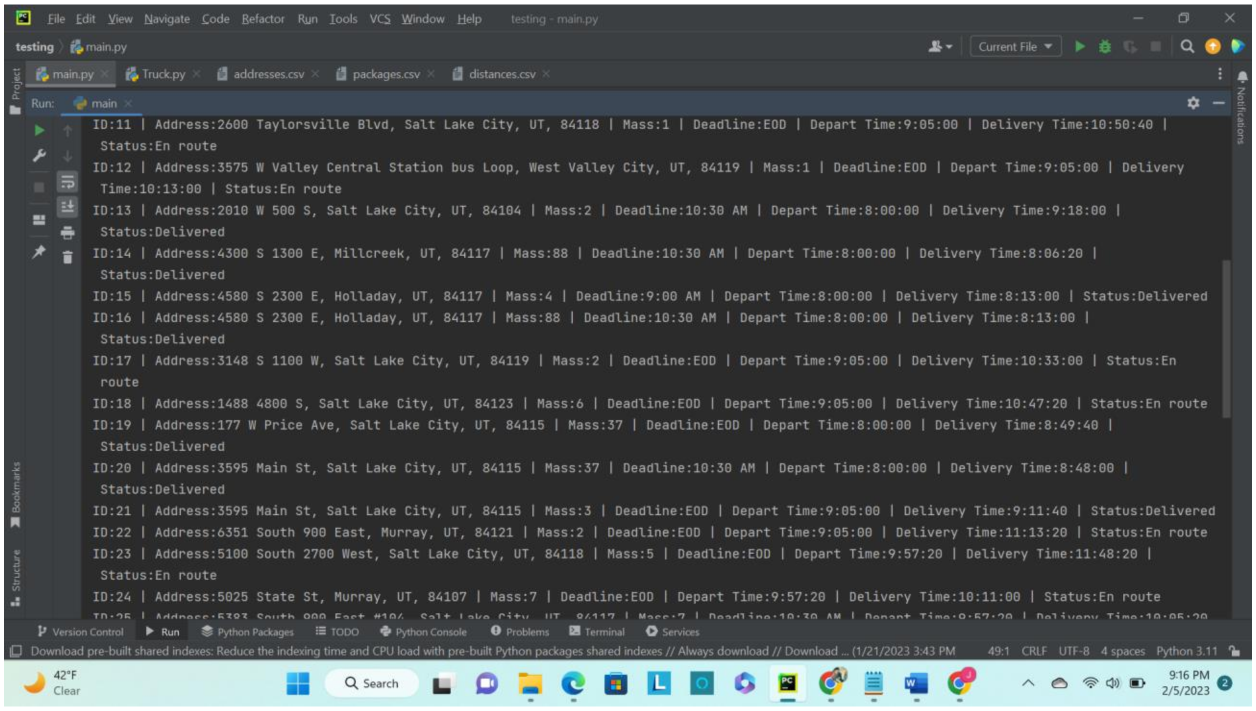
```
ID:27 | Address:1060 Dalton Ave S, Salt Lake City, UT, 84104 | Mass:5 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:54:20 | Status:At
  Hub
ID:28 | Address:2835 Main St, Salt Lake City, UT, 84115 | Mass:7 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:22:20 | Status:At Hub
ID:29 | Address:1330 2100 S, Salt Lake City, UT, 84106 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:29:40 |
 Status:Delivered
ID:30 | Address:300 State St, Salt Lake City, UT, 84103 | Mass:1 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:9:32:00 | Status:En
 route
ID:31 | Address:3365 S 900 W, Salt Lake City, UT, 84119 | Mass:1 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:58:40 |
 Status:Delivered
ID:32 | Address:3365 S 900 W, Salt Lake City, UT, 84119 | Mass:1 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:39:20 | Status:At Hub
ID:33 | Address:2530 S 500 E, Salt Lake City, UT, 84106 | Mass:1 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:26:00 | Status:At Hub
ID:34 | Address:4580 S 2300 E, Holladay, UT, 84117 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:13:00 | Status:Delivered
ID:35 | Address:1060 Dalton Ave S, Salt Lake City, UT, 84104 | Mass:88 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:54:20 |
 Status:At Hub
ID:36 | Address:2300 Parkway Blvd, West Valley City, UT, 84119 | Mass:88 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:23:20 |
 Status:At Hub
ID:37 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:9:28:40 |
 Status:En route
ID:38 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:9 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:43:40 | Status:At Hub
ID:39 | Address:2010 W 500 S, Salt Lake City, UT, 84104 | Mass:9 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:59:40 | Status:At Hub
ID:40 | Address:380 W 2880 S, Salt Lake City, UT, 84115 | Mass:45 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:42:40 |
 Status:Delivered
```

G2. Second Status Check

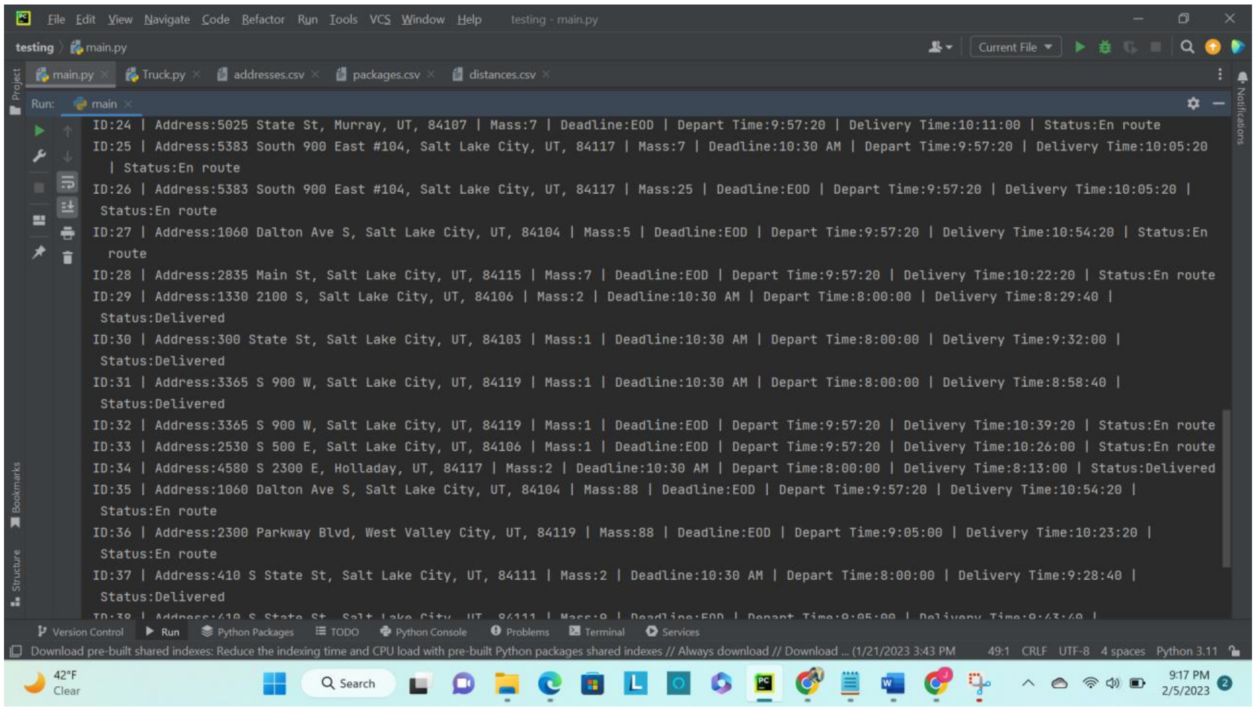
Package Information at 10:00 a.m.



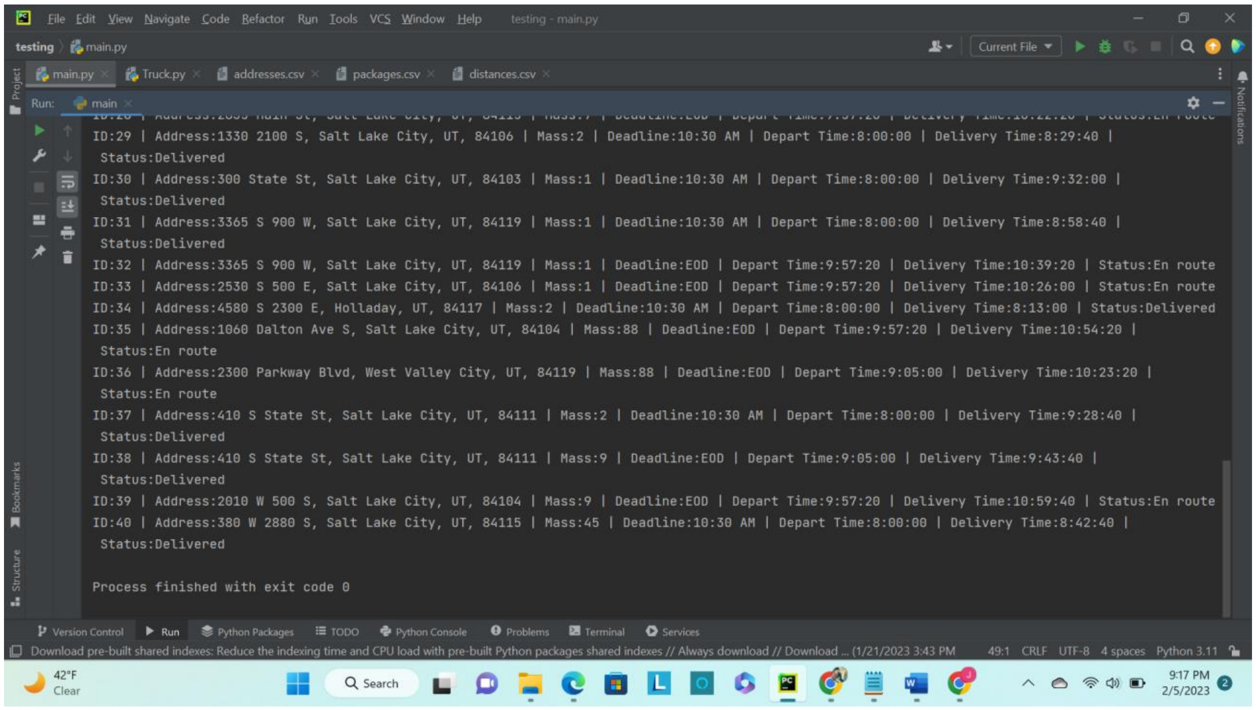
```
C:\Users\golay\AppData\Local\Programs\Python\Python311\python.exe "C:\Users\golay\Desktop\PyCharm Community Edition 2022.3.1\testing\main.py"
Welcome to Western Governors University Parcel Service
The total mileage for the delivery of these packages is: 107.0 miles
To start please type the word 'Start' (All else will cause the program to quit).Start
Please enter a time to search for the status of packages. Use the following format, HH:MM:SS10:00:00
Would you like to look up one package or all? Enter 'one' or 'all')all
ID:1 | Address:195 W Oakland Ave, Salt Lake City, UT, 84115 | Mass:21 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:39:00 |
 Status:Delivered
ID:2 | Address:2530 S 500 E, Salt Lake City, UT, 84106 | Mass:44 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:23:00 | Status:Delivered
ID:3 | Address:233 Canyon Rd, Salt Lake City, UT, 84103 | Mass:2 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:49:00 | Status:Delivered
ID:4 | Address:380 W 2880 S, Salt Lake City, UT, 84115 | Mass:4 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:17:00 | Status:Delivered
ID:5 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:5 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:43:40 |
 Status:Delivered
ID:6 | Address:3060 Lester St, West Valley City, UT, 84119 | Mass:88 | Deadline:10:30 AM | Depart Time:9:05:00 | Delivery Time:10:28:40 |
 Status:En route
ID:7 | Address:1330 2100 S, Salt Lake City, UT, 84106 | Mass:8 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:28:20 | Status:Delivered
ID:8 | Address:300 State St, Salt Lake City, UT, 84103 | Mass:9 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:47:00 | Status:Delivered
ID:9 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:2 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:11:10:20 | Status:En
 route
ID:10 | Address:600 E 900 South, Salt Lake City, UT, 84105 | Mass:1 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:37:40 |
 Status:Delivered
ID:11 | Address:2600 Taylorsville Blvd, Salt Lake City, UT, 84118 | Mass:1 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:50:40 |
 Status:En route
ID:12 | Address:3575 W Valley Central Station bus Loop, West Valley City, UT, 84119 | Mass:1 | Deadline:EOD | Depart Time:9:05:00 | Deli
```



```
ID:11 | Address:2600 Taylorsville Blvd, Salt Lake City, UT, 84118 | Mass:1 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:50:40 |
 Status:En route
ID:12 | Address:3575 W Valley Central Station bus Loop, West Valley City, UT, 84119 | Mass:1 | Deadline:EOD | Depart Time:9:05:00 | Delivery
 Time:10:13:00 | Status:En route
ID:13 | Address:2010 W 500 S, Salt Lake City, UT, 84104 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:9:18:00 |
 Status:Delivered
ID:14 | Address:4300 S 1300 E, Millcreek, UT, 84117 | Mass:88 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:06:20 |
 Status:Delivered
ID:15 | Address:4580 S 2300 E, Holladay, UT, 84117 | Mass:4 | Deadline:9:00 AM | Depart Time:8:00:00 | Delivery Time:8:13:00 | Status:Delivered
ID:16 | Address:4580 S 2300 E, Holladay, UT, 84117 | Mass:88 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:13:00 |
 Status:Delivered
ID:17 | Address:3148 S 1100 W, Salt Lake City, UT, 84119 | Mass:2 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:33:00 | Status:En
 route
ID:18 | Address:1488 4800 S, Salt Lake City, UT, 84123 | Mass:6 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:47:20 | Status:En route
ID:19 | Address:177 W Price Ave, Salt Lake City, UT, 84115 | Mass:37 | Deadline:EOD | Depart Time:8:00:00 | Delivery Time:8:49:40 |
 Status:Delivered
ID:20 | Address:3595 Main St, Salt Lake City, UT, 84115 | Mass:37 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:48:00 |
 Status:Delivered
ID:21 | Address:3595 Main St, Salt Lake City, UT, 84115 | Mass:3 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:11:40 | Status:Delivered
ID:22 | Address:6351 South 900 East, Murray, UT, 84121 | Mass:2 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:11:13:20 | Status:En route
ID:23 | Address:5100 South 2700 West, Salt Lake City, UT, 84118 | Mass:5 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:11:48:20 |
 Status:En route
ID:24 | Address:5025 State St, Murray, UT, 84107 | Mass:7 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:11:00 | Status:En route
ID:25 | Address:5383 South 900 East #104, Salt Lake City, UT, 84117 | Mass:7 | Deadline:10:30 AM | Depart Time:9:57:20 | Delivery Time:10:05:20
```



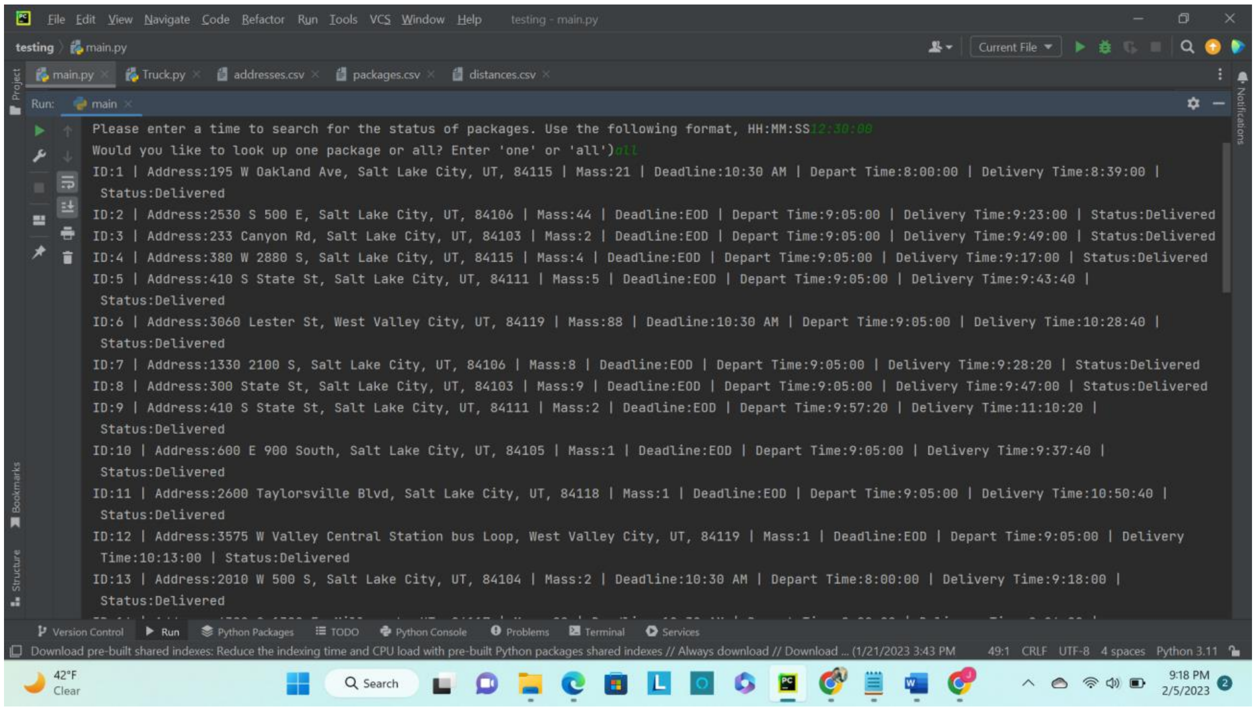
```
ID:24 | Address:5025 State St, Murray, UT, 84107 | Mass:7 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:11:00 | Status:En route
ID:25 | Address:5383 South 900 East #104, Salt Lake City, UT, 84117 | Mass:7 | Deadline:10:30 AM | Depart Time:9:57:20 | Delivery Time:10:05:20
  | Status:En route
ID:26 | Address:5383 South 900 East #104, Salt Lake City, UT, 84117 | Mass:25 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:05:20 |
 Status:En route
ID:27 | Address:1060 Dalton Ave S, Salt Lake City, UT, 84104 | Mass:5 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:54:20 | Status:En
  route
ID:28 | Address:2835 Main St, Salt Lake City, UT, 84115 | Mass:7 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:22:20 | Status:En route
ID:29 | Address:1330 2100 S, Salt Lake City, UT, 84106 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:29:40 |
 Status:Delivered
ID:30 | Address:300 State St, Salt Lake City, UT, 84103 | Mass:1 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:9:32:00 |
 Status:Delivered
ID:31 | Address:3365 S 900 W, Salt Lake City, UT, 84119 | Mass:1 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:58:40 |
 Status:Delivered
ID:32 | Address:3365 S 900 W, Salt Lake City, UT, 84119 | Mass:1 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:39:20 | Status:En route
ID:33 | Address:2530 S 500 E, Salt Lake City, UT, 84106 | Mass:1 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:26:00 | Status:En route
ID:34 | Address:4580 S 2300 E, Holladay, UT, 84117 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:13:00 | Status:Delivered
ID:35 | Address:1060 Dalton Ave S, Salt Lake City, UT, 84104 | Mass:88 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:54:20 |
 Status:En route
ID:36 | Address:2300 Parkway Blvd, West Valley City, UT, 84119 | Mass:88 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:23:20 |
 Status:En route
ID:37 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:9:28:40 |
 Status:Delivered
ID:38 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:9 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:43:40 |
```



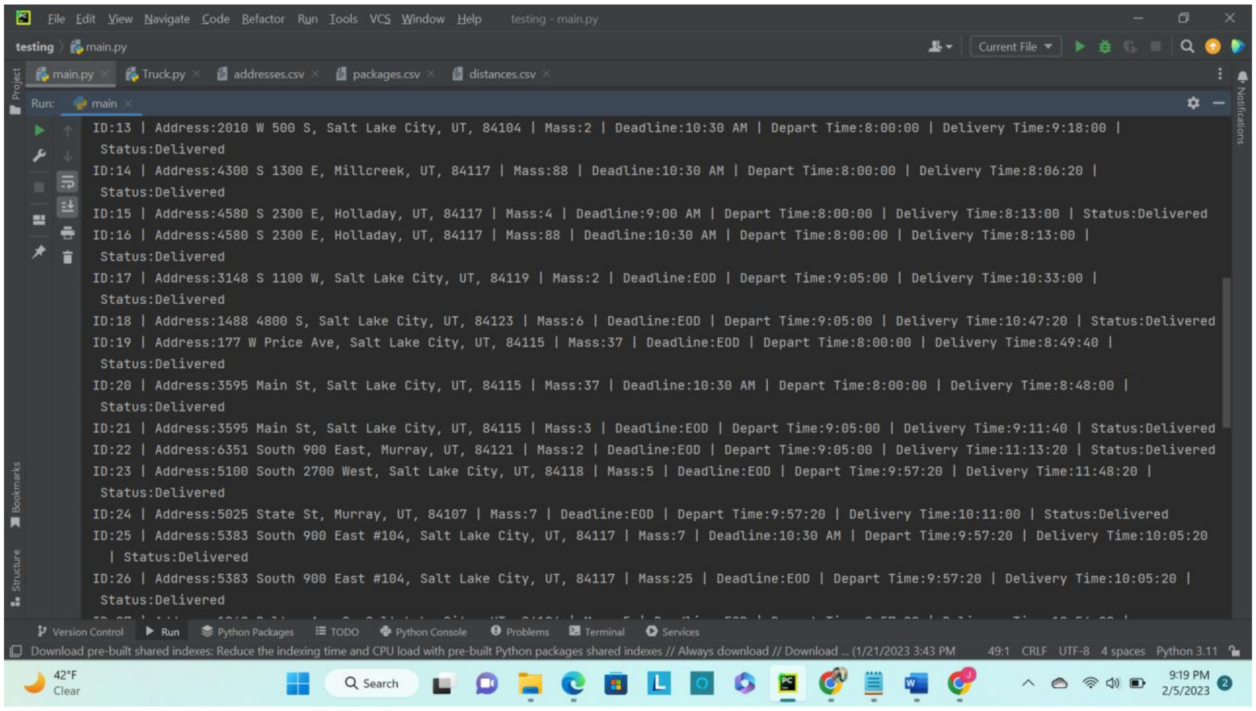
```
ID:29 | Address:1330 2100 S, Salt Lake City, UT, 84106 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:29:40 |
 Status:Delivered
ID:30 | Address:300 State St, Salt Lake City, UT, 84103 | Mass:1 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:9:32:00 |
 Status:Delivered
ID:31 | Address:3365 S 900 W, Salt Lake City, UT, 84119 | Mass:1 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:58:40 |
 Status:Delivered
ID:32 | Address:3365 S 900 W, Salt Lake City, UT, 84119 | Mass:1 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:39:20 | Status:En route
ID:33 | Address:2530 S 500 E, Salt Lake City, UT, 84106 | Mass:1 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:26:00 | Status:En route
ID:34 | Address:4580 S 2300 E, Holladay, UT, 84117 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:13:00 | Status:Delivered
ID:35 | Address:1060 Dalton Ave S, Salt Lake City, UT, 84104 | Mass:88 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:54:20 |
 Status:En route
ID:36 | Address:2300 Parkway Blvd, West Valley City, UT, 84119 | Mass:88 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:23:20 |
 Status:En route
ID:37 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:9:28:40 |
 Status:Delivered
ID:38 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:9 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:43:40 |
 Status:Delivered
ID:39 | Address:2010 W 500 S, Salt Lake City, UT, 84104 | Mass:9 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:59:40 | Status:En route
ID:40 | Address:380 W 2880 S, Salt Lake City, UT, 84115 | Mass:45 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:42:40 |
 Status:Delivered

Process finished with exit code 0
```

G3. Third Status Check

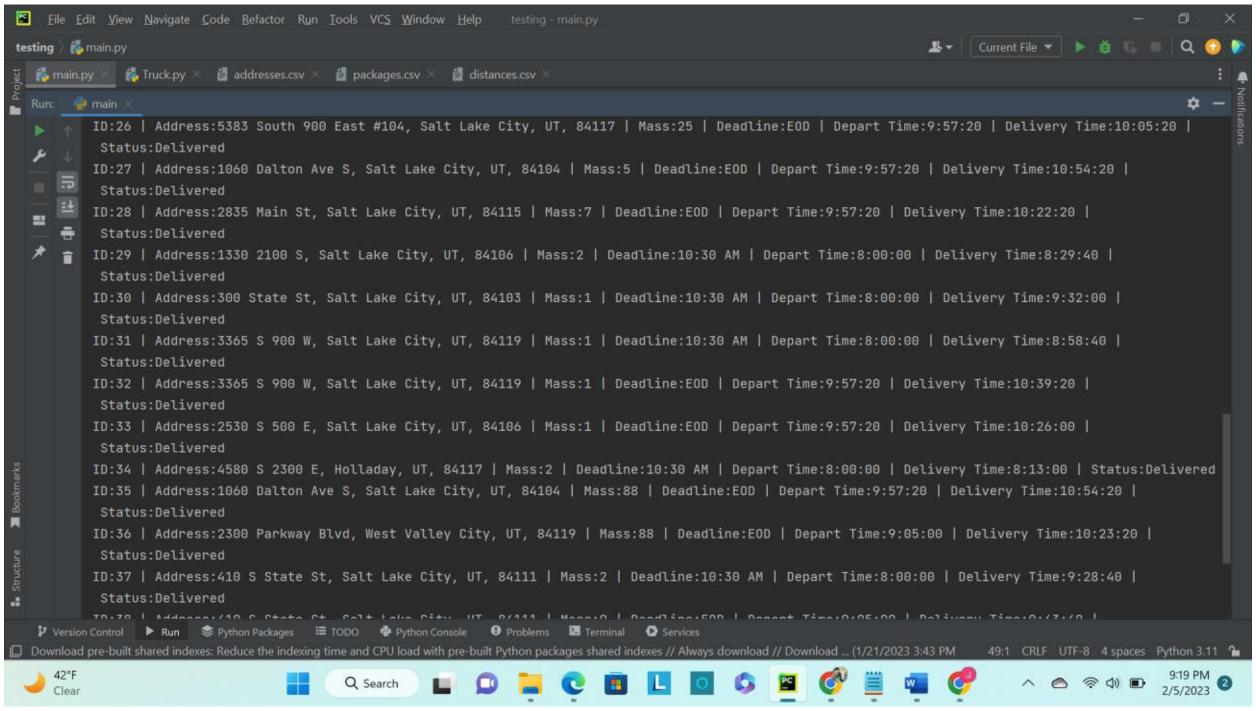
Package Information at 12:30 p.m.



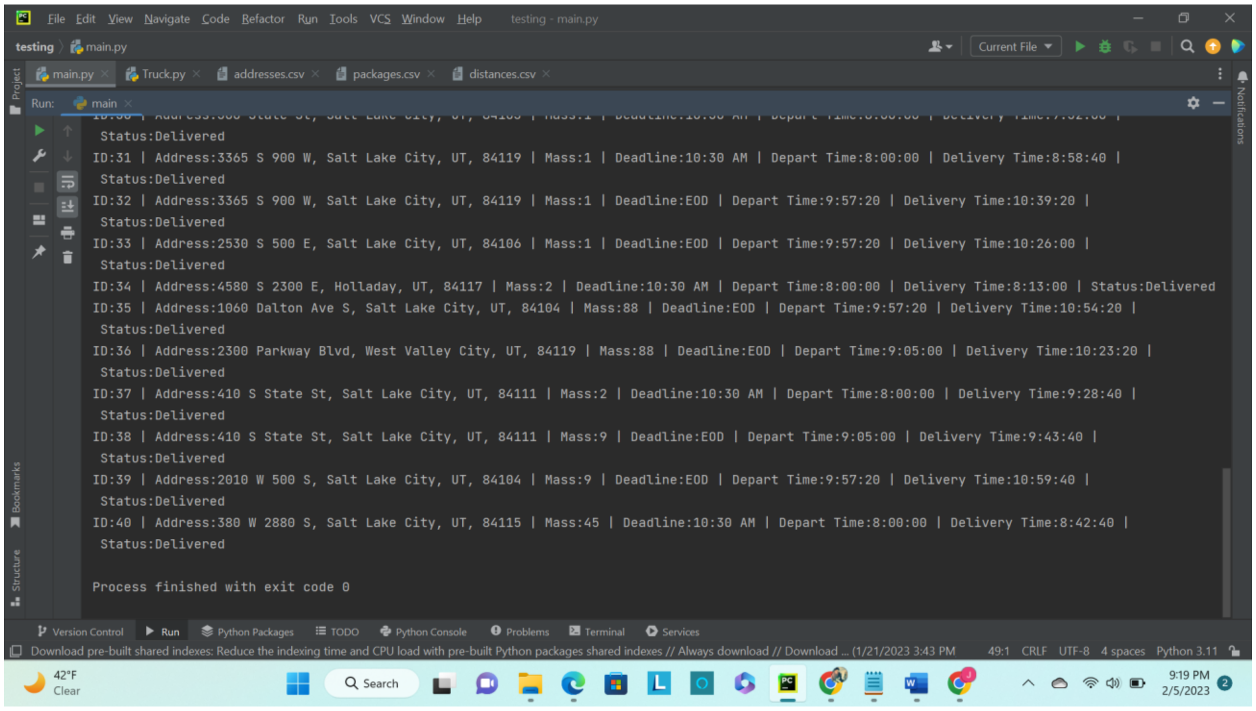
```
Please enter a time to search for the status of packages. Use the following format, HH:MM:SS12:30:00
Would you like to look up one package or all? Enter 'one' or 'all')all
ID:1 | Address:195 W Oakland Ave, Salt Lake City, UT, 84115 | Mass:21 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:39:00 |
 Status:Delivered
ID:2 | Address:2530 S 500 E, Salt Lake City, UT, 84106 | Mass:44 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:23:00 | Status:Delivered
ID:3 | Address:233 Canyon Rd, Salt Lake City, UT, 84103 | Mass:2 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:49:00 | Status:Delivered
ID:4 | Address:380 W 2880 S, Salt Lake City, UT, 84115 | Mass:4 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:17:00 | Status:Delivered
ID:5 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:5 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:43:40 |
 Status:Delivered
ID:6 | Address:3060 Lester St, West Valley City, UT, 84119 | Mass:88 | Deadline:10:30 AM | Depart Time:9:05:00 | Delivery Time:10:28:40 |
 Status:Delivered
ID:7 | Address:1330 2100 S, Salt Lake City, UT, 84106 | Mass:8 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:28:20 | Status:Delivered
ID:8 | Address:300 State St, Salt Lake City, UT, 84103 | Mass:9 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:47:00 | Status:Delivered
ID:9 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:2 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:11:10:20 |
 Status:Delivered
ID:10 | Address:600 E 900 South, Salt Lake City, UT, 84105 | Mass:1 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:37:40 |
 Status:Delivered
ID:11 | Address:2600 Taylorsville Blvd, Salt Lake City, UT, 84118 | Mass:1 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:50:40 |
 Status:Delivered
ID:12 | Address:3575 W Valley Central Station bus Loop, West Valley City, UT, 84119 | Mass:1 | Deadline:EOD | Depart Time:9:05:00 | Delivery
 Time:10:13:00 | Status:Delivered
ID:13 | Address:2010 W 500 S, Salt Lake City, UT, 84104 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:9:18:00 |
 Status:Delivered
```



```
ID:13 | Address:2010 W 500 S, Salt Lake City, UT, 84104 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:9:18:00 |
 Status:Delivered
ID:14 | Address:4300 S 1300 E, Millcreek, UT, 84117 | Mass:88 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:06:20 |
 Status:Delivered
ID:15 | Address:4580 S 2300 E, Holladay, UT, 84117 | Mass:4 | Deadline:9:00 AM | Depart Time:8:00:00 | Delivery Time:8:13:00 | Status:Delivered
ID:16 | Address:4580 S 2300 E, Holladay, UT, 84117 | Mass:88 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:13:00 |
 Status:Delivered
ID:17 | Address:3148 S 1100 W, Salt Lake City, UT, 84119 | Mass:2 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:33:00 |
 Status:Delivered
ID:18 | Address:1488 4800 S, Salt Lake City, UT, 84123 | Mass:6 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:47:20 | Status:Delivered
ID:19 | Address:177 W Price Ave, Salt Lake City, UT, 84115 | Mass:37 | Deadline:EOD | Depart Time:8:00:00 | Delivery Time:8:49:40 |
 Status:Delivered
ID:20 | Address:3595 Main St, Salt Lake City, UT, 84115 | Mass:37 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:48:00 |
 Status:Delivered
ID:21 | Address:3595 Main St, Salt Lake City, UT, 84115 | Mass:3 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:11:40 | Status:Delivered
ID:22 | Address:6351 South 900 East, Murray, UT, 84121 | Mass:2 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:11:13:20 | Status:Delivered
ID:23 | Address:5100 South 2700 West, Salt Lake City, UT, 84118 | Mass:5 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:11:48:20 |
 Status:Delivered
ID:24 | Address:5025 State St, Murray, UT, 84107 | Mass:7 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:11:00 | Status:Delivered
ID:25 | Address:5383 South 900 East #104, Salt Lake City, UT, 84117 | Mass:7 | Deadline:10:30 AM | Depart Time:9:57:20 | Delivery Time:10:05:20
  | Status:Delivered
ID:26 | Address:5383 South 900 East #104, Salt Lake City, UT, 84117 | Mass:25 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:05:20 |
 Status:Delivered
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help testing - main.py
testing main.py
Current File
main.py Truck.py addresses.csv packages.csv distances.csv
Run: main
Version Control Run Python Packages TODO Python Console Problems Terminal Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download ... (1/21/2023 3:43 PM 49:1 CRLF UTF-8 4 spaces Python 3.11
42°F Clear
Search
9:19 PM 2/5/2023

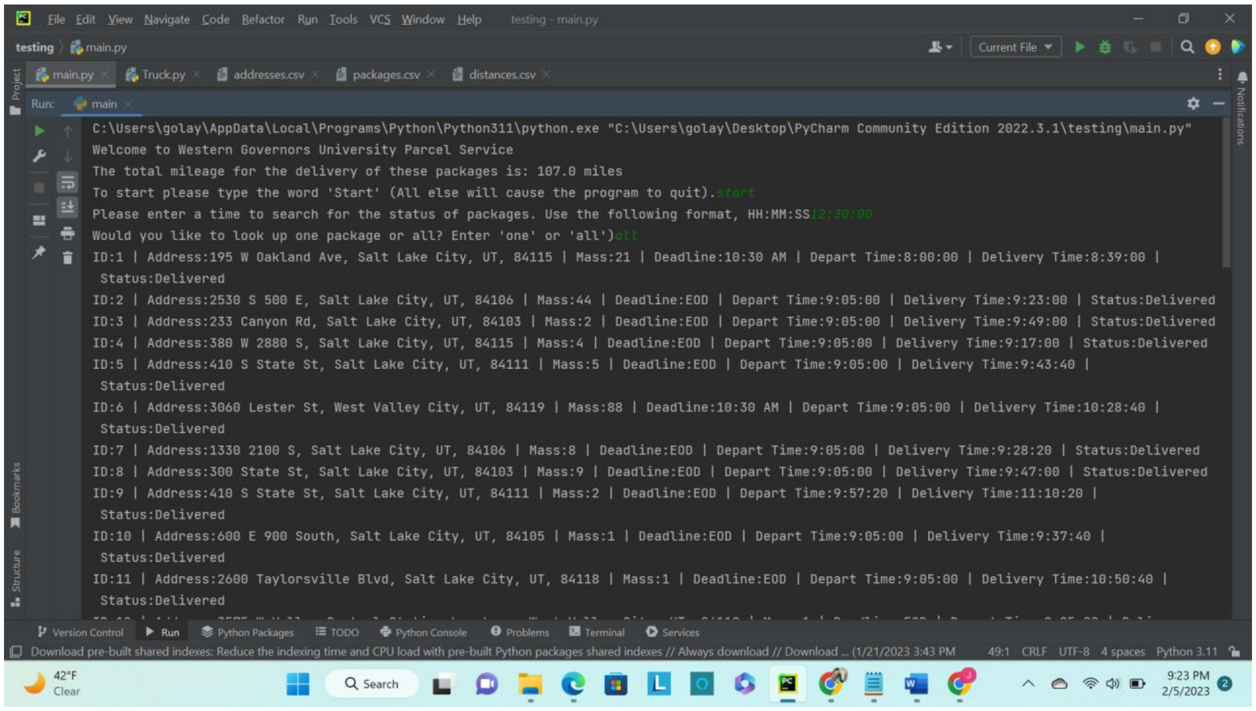
```
ID:26 | Address:5383 South 900 East #104, Salt Lake City, UT, 84117 | Mass:25 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:05:20 |
 Status:Delivered
ID:27 | Address:1060 Dalton Ave S, Salt Lake City, UT, 84104 | Mass:5 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:54:20 |
 Status:Delivered
ID:28 | Address:2835 Main St, Salt Lake City, UT, 84115 | Mass:7 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:22:20 |
 Status:Delivered
ID:29 | Address:1330 2100 S, Salt Lake City, UT, 84106 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:29:40 |
 Status:Delivered
ID:30 | Address:300 State St, Salt Lake City, UT, 84103 | Mass:1 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:9:32:00 |
 Status:Delivered
ID:31 | Address:3365 S 900 W, Salt Lake City, UT, 84119 | Mass:1 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:58:40 |
 Status:Delivered
ID:32 | Address:3365 S 900 W, Salt Lake City, UT, 84119 | Mass:1 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:39:20 |
 Status:Delivered
ID:33 | Address:2530 S 500 E, Salt Lake City, UT, 84106 | Mass:1 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:26:00 |
 Status:Delivered
ID:34 | Address:4580 S 2300 E, Holladay, UT, 84117 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:13:00 | Status:Delivered
ID:35 | Address:1060 Dalton Ave S, Salt Lake City, UT, 84104 | Mass:88 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:54:20 |
 Status:Delivered
ID:36 | Address:2300 Parkway Blvd, West Valley City, UT, 84119 | Mass:88 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:23:20 |
 Status:Delivered
ID:37 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:9:28:40 |
 Status:Delivered
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help testing - main.py
testing main.py
Current File
main.py Truck.py addresses.csv packages.csv distances.csv
Run: main
Version Control Run Python Packages TODO Python Console Problems Terminal Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download ... (1/21/2023 3:43 PM 49:1 CRLF UTF-8 4 spaces Python 3.11
42°F Clear
Search
9:19 PM 2/5/2023

```
 Status:Delivered
ID:31 | Address:3365 S 900 W, Salt Lake City, UT, 84119 | Mass:1 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:58:40 |
 Status:Delivered
ID:32 | Address:3365 S 900 W, Salt Lake City, UT, 84119 | Mass:1 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:39:20 |
 Status:Delivered
ID:33 | Address:2530 S 500 E, Salt Lake City, UT, 84106 | Mass:1 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:26:00 |
 Status:Delivered
ID:34 | Address:4580 S 2300 E, Holladay, UT, 84117 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:13:00 | Status:Delivered
ID:35 | Address:1060 Dalton Ave S, Salt Lake City, UT, 84104 | Mass:88 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:54:20 |
 Status:Delivered
ID:36 | Address:2300 Parkway Blvd, West Valley City, UT, 84119 | Mass:88 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:23:20 |
 Status:Delivered
ID:37 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:2 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:9:28:40 |
 Status:Delivered
ID:38 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:9 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:43:40 |
 Status:Delivered
ID:39 | Address:2010 W 500 S, Salt Lake City, UT, 84104 | Mass:9 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:10:59:40 |
 Status:Delivered
ID:40 | Address:380 W 2880 S, Salt Lake City, UT, 84115 | Mass:45 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:42:40 |
 Status:Delivered

Process finished with exit code 0
```

H. Screenshots of Code Execution

The code is executed free from runtime errors and warnings as evidenced by the screen shots provided above. The following screen shot shows the total mileage of all the trucks at the top of the user interface (107.0 miles):



```
C:\Users\golay\AppData\Local\Programs\Python\Python311\python.exe "C:\Users\golay\Desktop\PyCharm Community Edition 2022.3.1\testing\main.py"
Welcome to Western Governors University Parcel Service
The total mileage for the delivery of these packages is: 107.0 miles
To start please type the word 'Start' (All else will cause the program to quit).start
Please enter a time to search for the status of packages. Use the following format, HH:MM:SS12:30:00
Would you like to look up one package or all? Enter 'one' or 'all')all
ID:1 | Address:195 W Oakland Ave, Salt Lake City, UT, 84115 | Mass:21 | Deadline:10:30 AM | Depart Time:8:00:00 | Delivery Time:8:39:00 |
 Status:Delivered
ID:2 | Address:2530 S 500 E, Salt Lake City, UT, 84106 | Mass:44 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:23:00 | Status:Delivered
ID:3 | Address:233 Canyon Rd, Salt Lake City, UT, 84103 | Mass:2 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:49:00 | Status:Delivered
ID:4 | Address:380 W 2880 S, Salt Lake City, UT, 84115 | Mass:4 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:17:00 | Status:Delivered
ID:5 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:5 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:43:40 |
 Status:Delivered
ID:6 | Address:3060 Lester St, West Valley City, UT, 84119 | Mass:88 | Deadline:10:30 AM | Depart Time:9:05:00 | Delivery Time:10:28:40 |
 Status:Delivered
ID:7 | Address:1330 2100 S, Salt Lake City, UT, 84106 | Mass:8 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:28:20 | Status:Delivered
ID:8 | Address:300 State St, Salt Lake City, UT, 84103 | Mass:9 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:47:00 | Status:Delivered
ID:9 | Address:410 S State St, Salt Lake City, UT, 84111 | Mass:2 | Deadline:EOD | Depart Time:9:57:20 | Delivery Time:11:10:20 |
 Status:Delivered
ID:10 | Address:600 E 900 South, Salt Lake City, UT, 84105 | Mass:1 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:9:37:40 |
 Status:Delivered
ID:11 | Address:2600 Taylorsville Blvd, Salt Lake City, UT, 84118 | Mass:1 | Deadline:EOD | Depart Time:9:05:00 | Delivery Time:10:50:40 |
 Status:Delivered
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
testing - main.py
testing
main.py
Current File
main.py
Truck.py
addresses.csv
packages.csv
distances.csv
Run:
main
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download ... (1/21/2023 3:43 PM
49:1 CRLF UTF-8 4 spaces Python 3.11
42°F
Clear
Search
9:23 PM
2/5/2023

I1. Strengths of Chosen Algorithm

The Nearest Neighbor Algorithm has multiple strengths and was chosen primarily because of its time complexity, scalability, and ability to adapt to changes. The time complexity of the Nearest Neighbor Algorithm is O(N^2), which is significantly faster than the Brute Force approach to this problem, which requires a time complexity of O(N!). Furthermore, this algorithm can be scaled to larger inputs if needed for package delivery. Finally, this algorithm will execute as expected regardless of packages provided. When a delivery company must compute new routes each day, the algorithm can compute these routes so long as the addresses are correctly stored in the referenced lists.

I2. Verification of Algorithm

The program meets all requirements for this assignment. Trucks carry no more than 16 packages, trucks travel an average of 18 miles per hour, there are no collisions, drivers do not leave the hub earlier than 8:00 a.m., special instructions are followed for packages, and all packages are delivered. All data for packages can be looked up by package ID and time in the user interface, as required.

I3. Other possible Algorithms

A Brute Force Algorithm and Dijkstra's Algorithm ("Dijkstra's Algorithm") are both other possible algorithms that could be used to complete this assignment.

I3A. Algorithm Differences

The Brute Force Algorithm differs from the Nearest Neighbor Algorithm in both time complexity and outcome. Brute Force has a time complexity of O(N!) but does return the optimal route for package delivery. Alternatively, Nearest Neighbor has a time complexity of O(N^2) and returns an efficient route for delivery that may not be the most efficient possible in terms of distance travelled. The Brute Force approach would require calculation of every possible route to deliver the packages prior to choosing the most efficient, whereas Nearest Neighbor only calculates the nearest delivery address based on its current location and remaining packages.

Dijkstra's Algorithm utilizes a graph to find the shortest paths of delivering packages based on their distance from the delivery hub. The difference between this and Nearest Neighbor is that Dijkstra's will find the shortest path based on the distance from a delivery truck's starting point and Nearest Neighbor will update the shortest path based on its current location.

J. Different Approach

If I were to complete this project again following the same constraints, I would 1) adjust hashing function to be more complex and scalable, and 2) develop components of the project in more files for the purpose of simplification.

The hashing function currently utilizes modulo to hash package ID, however, other hashing approaches such as the Mid Square Method can be used to hash package IDs and maintain a fast look up rate with a larger number of packages because all of the digits in the ID assist in hash generation.

The program currently only has two programming files: main.py and Truck.py. Taking information from the package class and placing it into its own file would allow for more readable code.

K1. Verification of Data Structure

The hash table does meet the requirements for this project. It does not utilize a built-in python dictionary.

K1A. Efficiency

The time complexity of a look-up function slows if there are more than 40 packages or packages whose ID number hashes them into the same bucket. While an individual package in each hash bucket yields a time complexity of O(1), multiple items stored in the same hash bucket will result in a time complexity of O(N) as the look-up function must search through all IDs in that bucket.

K1B. Overhead

The hash table in this program is self-adjusting and grows with the amount of input it receives. The current program takes 40 packages and contains a list with 27 addresses. The hash table has a space complexity of O(N) and therefore will linearly require additional memory as additional packages and addresses are added. This could require additional servers.

K1C. Implications

Given this algorithm’s design, the number of trucks would not impact the look up time or space usage for this hash table. An increased number of cities would not increase the look up time as these are stored in a list and can be quickly referenced, however, the space usage would grow linearly.

K2. Other Data Structures

Two other data structures that could meet the same requirements for this assignment include a dictionary and an array.

K2a. Data Structure Differences

The hash table created in this program utilizes nested lists to store and locate items. Items are found using a search function. Alternatively, a dictionary stores key-value pairs and returns the value of an item given a key. It is possible that packages could be stored in an array for the purpose of this assignment, however, referencing items in an array would require knowledge if its index and would add complexity to correctly locating packages.

M. Professional Communication

Throughout this project and report, I have attempted to utilize professional communication for the purpose of explaining project requirements.

L. Sources - Works Cited

C950 - Webinar-1 - Let’s Go Hashing. Directed by Cemal Tepe, WGU,

http://wgu.hosted.panopto.com/Panopto/Pages/Auth/Login.aspx?instance=WGU&Auth=Se

ssionView&ReturnUrl=%2fPanopto%2fPages%2fViewer.aspx%3fid%3df08d7871-d57a-496e-a6a1-ac7601308c71.

Chase, Connor, et al. *An Evaluation of the Traveling Salesman Problem*. https://scholarworks.calstate.edu/downloads/xg94hr81q.

"Dijkstra's Algorithm." *Wikipedia*, 2 Feb. 2023. *Wikipedia*, https://en.wikipedia.org/w/index.php?title=Dijkstra%27s_algorithm&oldid=1137012547.

"Nearest Neighbour Algorithm." *Wikipedia*, 28 Feb. 2021. *Wikipedia*, https://en.wikipedia.org/w/index.php?title=Nearest_neighbour_algorithm&oldid=1009415297.