A. Identify a named self-adjusting algorithm (e.g., nearest neighbor algorithm, greedy algorithm) that could be used to create your program to deliver the packages.

The algorithm I have chosen for delivering the packages is the Greedy algorithm. Packages are delivered to the address closest to the truck's current address starting at the hub until all packages in the truck are delivered.

B. Identify a self-adjusting data structure, such as a hash table, that could be used with the algorithm identified in part A to store the package data.
   1. Explain how your data structure accounts for the relationship between the data components you are storing.

The data structure I chose was a Hash Table, which was used to hold all the packages in the system. Its initial capacity was ten and can be expanded.

C. Write an overview of your program in which you do the following:
C1. Explain the algorithm's logic using pseudocode.

Note: You may refer to the attached "Sample Core Algorithm Overview" to complete part C1.

```
While the number of packages in a truck > 0:

    //Finds the package in a truck that has an address closest to the truck's current address
    minDistance = Number that is bigger than all possible distances
    minPackage = Null
    For package in truck packages:
    If the distance between truck's current address and package's address < minDistance:
            minDistance = the distance between truck's current address and the package's address
            minPackage = package

    Truck's current address = minPackage's address
    The truck's current time += minDistance divided by 18 mph
    Package's delivery status = Delivered at truck's current time
    Remove this package from the truck
```

C2. Describe the programming environment you will use to create the Python application, including *both* the software and hardware you will use.

The software I am using for this project is PyCharm Community Edition 2022.2.1.The hardware I am using is a Lenovo ThinkPad X1 Yoga Gen 6 Laptop

C3. Evaluate the space-time complexity of *each* major segment of the program and the entire program using big-O notation.

C3a.main.py

distanceBetween(address1, address2):

O(1)

minDistanceFrom(fromAddress, truckPackages):

O(N)

truckDeliverPackages(truck):

O(N^2)

if __name__ == '__main__':

O(N^2)

C3b.CSV.py

loadPackageData(fileName, myHash):

O(N)

loadDistanceData(fileName, myList):

O(N^2)

loadAddressData(fileName, myList):

O(N)

C3c.Hash.py

__init__(self, initial_capacity=10):

O(N)

insert(self, key, item):

O(N)

search(self, key):

O(N)

remove(self, key):

O(N)

Since the space-time complexity never exceeds O(N^2) anywhere in the program, the space-time complexity for the entire program is therefore O(N^2).

C4. Explain the capability of your solution to scale and adapt to a growing number of packages.

I put all the packages inside of a Hash table, which is a chaining hash table composed of a three-dimensional list. Because the list data structure can have any number of new items added to it, the Hash table can therefore adapt to any number of packages as well.

C5. Discuss why the software design would be efficient and easy to maintain.

The software is easily maintainable because I have separated the different parts of it into files and was sure to well document it. If there is a need to update any part of the code, it is not difficult to navigate.

C6. Describe *both* the strengths and weaknesses of the self-adjusting data structure (e.g., the hash table).

Strengths: A hash table's main advantage is that searching (or inserting / removing) an item may require only O(1), in contrast to O(N) for searching a list or to O(log N) for binary search (Lysecky & Vahid 2018).

Weaknesses: As the number of items within the hash table increase so do the number of collisions.

C7. Key Justification

Package ID: each package's ID is unique thus making it the ideal key.

D. Sources

Lysecky, R., & Vahid, F. (2018, June). *C950: Data Structures and Algorithms II*. zyBooks. Retrieved March 22, 2021, from https://learn.zybooks.com/zybook/WGUC950AY20182019/