**Core Algorithm Overview**

**Stated problem:**

The purpose of this project is to find the best route for Western Governors University Parcel Service(WGUPS) packages and delivery distribution using Python(version 3.8.1) as the primary programming language. The main idea is to split 40 packages into 3 different "delivery truck" lists, yet only 2 drivers are available, which implies that only 2 trucks can be on the route at the same time, and when one of them is back in the hub, the last one can start delivering. Also, some packages have a delivery deadline, which should be delivered on time and the package with the wrong address listed, and it will only be updated at 10:20 am.

**Algorithm:**

The algorithm of choice is a greedy algorithm with slight variation, where it looks for the packages closer to the route and sorts it by the distance from one another. It meets the requirements of the project with total delivery mileage within 111.3 miles. The first advantage is its ability to discover the quickest route with the package list given. Second, it is capable of handling larger inputs.

**B2. Application of Programming Models**

First, the application reads the data from CSV files and creates the Package and Location objects. To track the distances between locations, it empowers the Graph data structure. Then it sorts the packages to "early delivery" list, "truck 2" list, and "wrong address" list. Next, it applies a Greedy algorithm to distribute the packages by the distance among them. If the number of packages is less then the truck capacity, the algorithm looks for additional packages which location is in the short distance from the main route. To operate the program, the user enters choices into the console user interface to display data at the time specified by the user. Additionally, the application does not utilize the communicational protocol, as everything operates on a local machine.

**B4. Adaptability**

The core functionality of the application was designed with scaling and the ability to handle more packages in mind. For instance, methods that handle reading in the data from the CSV files can process more input data, which implies more locations and packages. However, a similar technique will be applied to determine the shortest route. One of the downsides of the functionality is the hardcoded number of trucks. Which again can be later modified to accommodate the growth of the business.

**B5. Software Efficiency and Maintainability**

The application is quite efficient with polynomial-time complexity O(N^2). Again, it can be improved by finding a better, more efficient approach to the problem. For maintainability, the project filled with comments, which offers more comfortable to read and follow the logic of a particular piece of the functionality. Furthermore, the project implements a "single responsibility principle" for instance, classes separated into different files, such as Time, Truck, Package, etc., which provides the capability to adapt or update.

**B6. Self-adjusting data structures**

The primary data structure that was used in the project is the custom hash table. The main advantage is the constant time complexity O(1), which is quite efficient. Another data structure I used for the delivery truck route is the queue(FIFO), which is also quite efficient with the constant time complexity O(1). Both data structures can grow subsequently to accommodate more data to input while keeping constant time complexity.