**Sample Core Algorithm Overview**

**Stated Problem:**

The purpose of this project is to create an algorithm using the Python programming language to develop an effective and efficient solution for the traveling salesman problem (TSP).

This assignment adds additional requirements of:

* Package delivery deadlines
* Packages which must be on the same truck
* Packages which must be on a specific truck
* Truck load limitation
* Truck speed constraint
* 2 Trucks can be operational at one time.
* One package has the wrong address assigned and must be fixed at 10:20am

Program Blueprint and architecture

**Algorithm Overview:**

The proposed solution for this assignment uses a brute force method which tried to insert a delivery

Using brute force

The Traveling Salesman problem

Delivery Chart of which truck each delivery was assigned

**Text

Description automatically generated**

Route table listing the locations each specified truck will go through throughout the day.

**Text

Description automatically generated**

Package table of each package and their equivalent Truck assigned and Limitation Notes, Deadline to arrive.

**Text

Description automatically generated**

Truck table notifying the current location of the truck, packages on board and miles it will travel during the assigned route.

**Text

Description automatically generated**

**A: Algorithm Selection**

Brute Force Algorithm

**B1: Logic Comments**

**B2: Application of Programming Models**

**B3: Space-Time and Big-O**

**B4: Adaptability**

**B5: Software Efficiency and Maintainability**

**B6: Self-adjusting Data Structures**

**C: Original Code**

**C1: Identification Information**

**C2: Process and Flow Comments**

**D: Data Structure**

**D1: Explanation of Data Structure**

**E: Hash Table**

**F: Look-up Function**

**G: Interface**

**G1: First Status Check**

**G2: Second Status Check**

**G3: Third Status Check**

**H: Screenshots of Code Execution**

**I1: Strengths of the Chosen Algorithm**

**I2: Verification of Algorithm**

**I3: Other Possible Algorithms**

**I3A: Algorithm Differences**

**J: Different Approach**

**K1: Verification of Data Structure**

**K1A: Efficiency**

**K1B: Overhead**

**K1C: Implications**

**K2: Other Data Structures**

**K2A: Data Structure Differences**