**Core Algorithm Overview**

**Algorithm Overview:**

The KNN (K Nearest Neighbor) algorithm is created in the following manner:

def deliver\_packages(package\_hashtable, truck):  
 time = truck.departure\_time  
 current\_address = 0  
 removal\_index = 0  
 new\_address = 0  
 shortest\_distance = 5000.0  
 while truck.package\_list:  
 for package in truck.package\_list:  
 dist\_between = float(distance\_between(current\_address, address\_data.index(package.address)))  
 if dist\_between <= float(shortest\_distance):  
 shortest\_distance = dist\_between  
 removal\_index = truck.package\_list.index(package)  
 new\_address = address\_data.index(package.address)  
 current\_address = new\_address  
 removal\_id = truck.package\_list[removal\_index].package\_id  
 package\_hashtable.search(removal\_id).package\_status = 'Delivered'  
 truck.package\_list.remove(truck.package\_list[removal\_index])  
 miles\_to\_minutes = shortest\_distance \* 60 / 18  
 time += timedelta(minutes=miles\_to\_minutes)  
 package\_hashtable.search(removal\_id).time\_delivered = time  
 truck.miles\_traveled += shortest\_distance  
 shortest\_distance = 5000  
 final\_address = current\_address  
 truck.miles\_traveled += float(distance\_between(final\_address, 0))

Step one is to create the variables that will be constantly updating as the algorithm iterates over the truck’s packages and delivers them.

Variables: A time variable is created to keep track of time. Current\_address is created to hold the location currently visited. Removal\_index is created for the purpose of grabbing the current packages index for removal. New\_address is for storing the new address index. Shortest distance is created and initialized to a high number and will update every iteration that contains a shorter distance.

While there is an item in the truck,

For every package in the truck, take the distance between the current address index and the current package’s address.

If the distance between is lower than the current value of the shortest distance variable, update the shortest distance variable to the distance between.

Take the index of the current package and store it in the removal index variable

Update new address to the index of the current package address.

after iterating through the package list in the truck, store the value of new address in the current address. This is now the location we are visiting.

Create and initialize the variable removal\_id to store the package ID and remove the package from the truck containing that id.

Search the hashtable using the removal\_id as a key and update the package\_status to “Delivered”

Remove the package with the removal id from the truck’s package list.

Convert the miles traveled to minutes using the formula distance \* 60 / 18

Add the time to the time variable that stores the current time

Update the package’s delivery time in the hash table

Add the shortest distance to the truck’s miles traveled

Reset the shortest distance for the next iteration

Return the truck to the hub by getting the distance between the final address and the hub and adding them to the miles traveled.