C950 Algorithm Write Up (Part 2)

Cameron S Minty

F. Justify the package delivery algorithm used in the solution as written in the original program

1. The ‘Nearest Neighbor” algorithm has the following strengths and weaknesses.
   1. Strengths: The nearest neighbor algorithm is extremely simple to implement and understand. Additionally as a “lazy learning” algorithm it does not require a training phase and is therefore quick to implement.
   2. Weaknesses: Nearest neighbor has poor performance with noisy data as it does not account for spread or clustering, this can lead to inefficient hopping between addresses. The algorithm also has a relatively poor performance with O(N) per query performance.
2. The nearest neighbor algorithm successfully delivers all packages on time and also accounts for specific package situations such as the delayed or grouped packages by using some tweaking. It meets the requirements of the scenario.
3. Two other algorithms that would meet the requirements.
   1. KNN (K-Nearest Neighbor)
      1. KNN is a broader implementation of the nearest neighbor algorithm that adds the hyperparameter ‘K’ that represents a cluster of nearest neighbors. This accounts for noisy data by adding higher processing overhead. KNN will select a nearest neighbor by either voting amongst possible neighbors or selecting a mean distance.
   2. Djikstra’s Algorithm
      1. Djikstra’s algorithm, most famously used in the OSPF routing protocol, is used to find the shortest path between any two points. It works by building a weighted graph based on ‘distance’ and then uses that to plan an optimal path from one node to the next. It is a ‘greedy’ algorithm meaning that it will always take the shortest distance no matter what.

G. Describe what you would do differently other than the two algorithms listed above.

1. I would implement some kind of standardized way of handling the package notes. With a larger package data set more general patterns could be found for the format of package notes and what to do with packages with different kinds of notes could be standardized and worked into the package loading algorithm without hard coding loading decisions.
2. A web based user interface could be implemented and would allow for a user to have a better UX when viewing the package data. This could be implemented in Flask for simplicity and portability
3. For scalability it may be good to use some kind of database software to store time stamped state changes. As it stands now all data regarding packages is stored in main memory but as the scale and complexity of the program increases it will become untenable to store all of the program in main memory.

H. Verify that the data structure used in the solution meets all requirements

The data structure selected, Hash Table, allows for O(1) retrieval of elements and O(N) search and removal. The data structure is used to keep track of the status of all current packages by creating a “package interface” object that can be used to easily store the stats associated with any specific package. The HashTable allows for inserting packages by ID, retrieving and removing them by ID and therefore meets all requirements.

1. Heap Queue:
   1. A heap queue is a data structure that can be used to create priority heaps. In a heap data structure each element is ‘lower’ than the element to its left, this can be used to create a data structure that will select for the lowest value element at the top. In a heap queue retrieval of the lowest value (distance) object is O(1) time and searching/removing is O(logN). The heap queue could be used to achieve the “greedy” organization of the packages that is being implemented through sorting by the nearest deadline.
2. An Array
   1. Any array is a linear data structure that maps elements to indexes. Arrays are the basis for the hash table but more primitive. Packages have a clear integer package\_id value that could be used as an analogue for an array index. By building an array with length *n* where *n* is the largest package\_id in the inventory we can still achieve the O(1) direct access. In order to prevent dynamic array resizing either a fixed array could be used or when packages are removed then their respective index could be set to a void pointer (None value).