Task 1 Written Submission 1

By: Nyla Gaston

Date: 8/24/2023

1. **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 Greedy algorithm is a self-adjusting algorithm that could be used during the creation of my program and utilized to deliver the packages.

1. **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.**

A hash table is a self-adjusting data structure that could be used in conjunction with the algorithm in order 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 that I used for my program was a hash table which makes data components easily accessible and easy to retrieve. A chaining hash table places keys into buckets, with each bucket only holding one key and instead of one single value, the value is a linked list which will prevent collisions and the program crashing. The hash table I used was a chaining hash table which has benefits such as efficient search, removal, and insertion of the data components, and an overall reduction of data collisions. With the hash table for the packages, the package ID data component was used as the key for the hash table with the value being a list of other package details such as status, weight, and any special notes. The key is the package ID and the values being an array of data components means that all data components are associated with that corresponding key and to each other being stored in the buckets in the hash table.

**C.  Write an overview of your program in which you do the following:**

**1.  Explain the algorithm’s logic using pseudocode.**

**Nearest Neighbor Algorithm**

**PackagestoDeliverList = (1, 2, …, 40)**

**While PackagestoDeliverList > 0**

**Find next package to deliver**

**If PackageID = 6, 25**

**PRIORITIZE**

**If not**

**NextPackage = closest address from current location**

**After NextPackage is delivered**

**Remove from PackagestoDeliverList**

**MilesDriven = distance driven to each address**

**Time = MilesDriven/ 18MPH**

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

The programming environment I will use starting with the hardware is a 2017 MacBook Pro running MacOS Ventura for the operating system. The MacBook is connected to an external monitor using an HDMI cord and connected via Bluetooth to an external keyboard and mouse.

The software environment for this program consists of Visual Studio Code 3 to actually write the program in Python using version 3.11.4 and again MacOS Ventura as an operating system for my computer.

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

The space-time complexity of each major component of the program is described below. The hash table has a notation of O(1). The nearest neighbor algorithm has a complexity of O(n^2) this is because it is a nested loop and dependent on size this is also the complexity of the function that loads the data from the package CSV into the algorithm and the function that finds the minimum distance between addresses as it depends on the size of the list and contains nested loops. The function that finds the distance between two given addresses is a complexity of O(n) as it is linear and only depends on the size of the list, going through it once. The print all function has a complexity of O(n) as it must go through the entire list as the print by package ID function is O(1) as it simply searches the hash table by the package ID. Overall the program has a space-time complexity of O(n^2).

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

My programming solution is extremely scalable and customizable to suit a growing number of packages. To start the package class ensures that all packages will have the necessary information to be run through the algorithm and will not accept packages that don’t have all the required information. Secondly, there is the option to add more trucks using the truck class which will allow more packages to go out and deliver more packages. Lastly, the functions and algorithm will continue to work at the same speed independently of the number of packages or trucks, so adding either will not reduce performance and will be very easy to do.

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

The design of the software would be easy and efficient to maintain because first of all there aren’t many parts to it therefore if something needed to be changed, it wouldn’t be difficult to change all that needs to be changed because it’s very easy to navigate. The second reason it’s easy to maintain is that the code is written very efficiently, as there are classes and definitions and there’s order, not code and methods sprayed all over the code so if something goes wrong it can quickly be pinpointed within the code and fixed. The program is efficient because it utilizes a great data structure and algorithm for the situation. The hash table data structure keeps the time at O(1) which is the most efficient that a data structure can be. This program ensures that minimal time or resources are used while producing maximum results and not risking performance. This pairing of the hash table and nearest neighbor algorithm, compared to say a doubly linked list and Dijkstra’s algorithm and other pairings, produces faster results for the program while not decreasing the program’s productivity.

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

The strengths of the hash table are that they are very fast, very simple and easy to implement, and great for arrays. Another strength is that “chaining a is a collision resolution technique where each bucket has a list of items” (Lysecky, R., & Vahid, F., 2018). There are also weaknesses for the hash table in that it does use extra storage as it requires a few empty slots in order to remain efficient.

**7.  Justify the choice of a key for efficient delivery management from the following components:**

The package ID is the most efficient choice for a key for the purpose of efficient delivery management. The reason this is the best choice is that first of all many packages could have the same delivery address, city, zip, weight, status, or deadline. These data components are not unique enough to make effective and efficient keys and could cause problems in getting packages to the right place and on time. The uniqueness of the package ID makes sure the correct package information is stored as the value to the matching key and therefore ensures an efficient algorithm and data structure can be built using it.

**D.  Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.**

Lysecky, R., & Vahid, F. (2018, June). *C950: Data Structures and Algorithms II*. zyBooks.

Retrieved March 22, 2021, from <https://learn.zybooks.com/zybook/WGUC950AY20182019/>