Author: Caleb Ewer

Date: 11/24/23

ID: 2409827

CS 260 Module 5 Journal

For the given task we have been working on (the ebids) we are faced with the challenge this week to use a hash table to complete it. Just like with past assignments we had most of the main class completed and we had to work on implementing the other methods and structures to create the algorithm. In our version of the hash table, we used the method that in the event of a collision we will simply add it to the location via a linked list.

The hash table is good for solving the problem because it sorts the files by their bid id and arranges them by dividing by a modulus of the id and stores them in that. For the first smaller file it works much better because there are only about 180 or so entries which make the time to complete them much easier. But when it comes to the larger file, we tend to have several more double, triple, and even larger used locations used to store the elements. Because of the larger overused locations of linked lists, finding the element in the linked list can be difficult and time consuming. In this case of the hash table we can most likely find a better use of a pair of algorithm and data structure but if we are patient we can use the hash table to complete our task.

The other uses that I feel something like this could be used is in an inventory management system. In my current job we use a system that is similar for our racking. In that racking system we have a double deep location for product and it is arranged by skus, using a hash map would allow us to be able to manage the locations a little easier. Another use could be to use for a distribution system like a server. We could be able to use a key of people so that when we have too many people using one we could move them into a different one.

Ultimately there are many more purposes we could use this kind of a system for and in our purpose it gets the job done but with time being a small issue. Maybe choosing to search for an empty location would have been a better choice than to put them in a linked list.