Grant Mercer

Assignment 8

Part B

**Summary of hash table**

The hash table is commonly an array of some fixed size containing items. Each item is kept track of by a key mapped by a hash function. The function ideally ensures that no two keys overlap. This allows large amount of data to be stored in fixed size containers regardless of the data type. This is especially useful as the user can now take advantage of random access lookup and contingent memory storage.

**Comparison of hash table and binary search tree**

A hash table can offer some significant advantages over a binary search tree. A binary search tree is not contingent in memory and therefore does not have random access, making lookup’s O(log n). A hash table has a fixed container, and in best case with no collisions has a constant lookup time O(1). On the other hand a binary search tree has a guaranteed worst case insert of O(log n). If a hash table is plagued with collisions the insertion of a piece of data could worst case take O(n) with an inefficient collision solver.

**Explain why the instructors name was added first and showed near the end**

The reason the instructors name appears near the last during the print statement is that the Modulo of the instructors last 4 digits and the tablesize is a large number, while the Heinz last 4 digits result in a value of zero after the hash. This makes Mr. Jorgensons contact placed near the back of the container.

**Note why initial table size was 503 and explain what occurs when the load factor is reached**

The initial size of the table was set to 503 to account for last 4 digits of the entered phone number and still avoid collisions as much as possible. As the load factor reaches 0.65, the amount of collisions is so great that the insertion of data begins getting quite slow and requires the table to be enlarged to avoid too many collisions.

**Explain why hash function used is good or poor**

The hash function used for this application was sufficient and good to use for the set of data. A simple modulo of a 4 digit number would allow a result in a small enough range that a fixed container could be created and managed without too much enlarging or fear of collisions.

**Big-O for various hash operations**

Insert: O(n)

Search: O(1)

Rehash: O(n)