Zakeriya Muhumed
Program #3 Writeup
CS163

Hash tables are an effective tool that allow for quick data insertion, deletion, and retrieval. I had to build a hash table from scratch for this programming project in order to store and organize a group of data.

I have to first grasp the fundamental ideas behind hash tables. A hash function is used by a hash table, a type of data structure, to convert a given key into an array index. Once a key has been assigned a mapping, the index value can be used to insert or retrieve the data from the array. To prevent collisions—where two keys map to the same index value—the hash function must have a uniform distribution of keys throughout the array.

I started by developing a hash table class to control data storing and retrieval. Setting the array's size, which would house the data, was the initial step. I made a size choice based on the anticipated volume of data to be kept. I then created a hash function that would convert a key into an array index. I calculated the hash value using the modulo operator, which results in a consistent distribution of keys throughout the array.

I used the distinct chaining strategy to handle collisions, which includes keeping all information that hashes to the same index in a linked list. To handle the collisions, I developed a linked list class, and each element of the array had a pointer to the head of the matching linked list.

Once the fundamental data structure was in place, I added public member methods to let data be added, found, and removed from the hash table. The hash function was utilized by the insert function to determine the array's index after accepting a key and data value. A new linked list was built to store the data if there was no existing linked list for that index. If not, the linked list at that index received the data.

Once the fundamental data structure was in place, I added public member methods to let data be added, found, and removed from the hash table. The hash function was utilized by the insert function to determine the array's index after accepting a key and data value. A new linked list was built to store the data if there was no existing linked list for that index. If not, the linked list at that index received the data.

Using the key, the delete function deleted data from the hash table. It utilized the hash function to calculate the array's index and then searched the linked list at that index for the data to be

eliminated. This is similar to how the search function works. If the information was located, it was taken out of the linked list. The linked list was erased to free up memory if it was empty after the data had been removed.

I had to evaluate the hash table's performance once I had created the fundamental data structure and the public member functions. In order to accomplish this, I developed a test script that added a variety of key-value combinations to the hash table, searched for and retrieved data from the hash table, and deleted data from the hash database. The test script timed how long it took to accomplish these operations for various input sizes in order to gauge the performance of the hash table.

Overall, this task was gratifying but tough. I gained a lot of knowledge about hash tables, their operation, and how to use them in real-world applications. I was able to improve my programming abilities through this project by learning how to build a reliable hash table that can handle a variety of inputs. I was also able to use this knowledge in practical contexts, which made it easier for me to understand the usefulness of data structures like hash tables.