Hashing Function and Hashmap Implementation

Homework #9

By

John Kiyak

CS303

October 30, 2018

**Problem Specification**

The goal for this assignment was to implement a hashing function, use the hashing function to create a hashmap implementation, and implement linear and quadratic probing methods and compare its performance with Red and Black Trees.

**Program Design**

For this program, the main function reads and stores data from ‘input.dat’ and ‘UPC.csv’ into separate arrays. Then, a key is chosen using the random.choice function, from the ‘input.dat’ file, and store into the key variable. Then, the key is printed. The insert function insert the chosen variable into the node. The inorder class prints out the tree if it is in order. The search function takes in the root of the tree, and the key that is being looked for, and returns either the root, or continues searching until the key is found in the root.

**Testing Plan**

For my testing plan, I would make sure that the arrays for both the ‘input.dat’ and ‘UPC.csv’ files were printed and stored correctly. I would make sure the user can both put and get data to and from the hashmap. Then, I would make sure the linear and quadratic probing implementations were working as intended. Finally, I would make sure the UPC.csv files and in the input.dat files were being read and that the keys were being put into the hashmap and correctly working, and compare the time taken for the keys in the hashmap, linear, and quadratic probing mechanisms.

**Analysis and Conclusions**

According to Donald Kruth, “when using linear probing, dictionary operations can be implemented in constant expected time. This means that when inserting, removing and search operations in a hash table, it has a time complexity of O(1) when linear probing is used. Going into more detail, the time for any particular operation in linear probing is proportional to the length of the contiguous block of occupied cells at which the operation starts.”

For quadratic probing however, There is no guarantee of finding an empty cell once the table gets more than half full, if the table size is not prime. Since quadratic probing is only useful for prime numbers in the modulus, linear probing seems like the best bet in terms of hashing.