COLORADO MESA UNIVERSITY

Part A Answers

By Drake Cullen

I declare that all material in this assessment task is my work except where there is clear acknowledgement or reference to the work of others. I further declare that I have complied and agreed to the CMU Academic Integrity Policy at the University website. http://www.coloradomesa.edu/student-services/documents

Author's Name: Drake Cullen **UID(700#):** 700480375 **Date:** 10/17/2021

Statement a)

I tested my hashtable with occupancy ratios of 50%, 70%, and 80%. For each occupancy ratio, 7,650,000 lookups were performed with random words so that a discernable time difference could be found.
The occupancy ratio of 50% took 1,072,580 clock cycles and 1.07223 seconds to perform the lookups. There were 1,650 collisions.
The occupancy ratio of 70% took 1,091,944 clock cycles and 1.09161 seconds to perform the lookups. There were 1,898 collisions.
The occupancy ratio of 80% took 1,0122,989 clock cycles and 1.12265 seconds to perform the lookups. There were 2,049 collisions.
I decided to use an occupancy ratio of 50% for the rest of my experiments.

Statement b)

I used open hashing, so my hash table could handle variable sizes by adding new elements to the linked list.

Statement c)

I started off with the hash function f(r) = r % hsize. Next, I incorporated the following hash function from our slides:

```
int hash(char * key)
{
   int val = 0;
   while(*key != '\0')
   {
    val = (val << 4) + (*key);
    key++;
   }
   return val;
}
```

In order to minimize collisions, I created a set of nested loops to change the initial val to numbers in the range from 0 to 10,000. Every time I changed the initial value, the bit shifting variable looped through values of 0 to 25. In the end, I found an initial value of 530 with a left shift of 8 minimized the number of collisions.

Statement d)

If there is a collision, I iterate over the corresponding linked list and insert the new word in alphabetical order.

Statement e)

No, you don't need an interface class for each function. I put my hashtable in a class, so I did use an interface class for the hashtable methods.