Stephen Townsend
CS 261
3/12/17

Assignment 6 Written Questions

1. Give an example of two words that would hash to the same value using hashFunction1 but would not using hashFunction2.

Battle and Cattle. Hash function2 would not work because it accounts for the ordering if the words.

2. Why does the above observation make hashFunction2 superior to hashFunction1?

Hashfunction2 is superior because it allows ordering of the characters and thus in turn less links would be needed I the hash table.

3. When you run your program on the same input file once with hashFunction1 and once with hashFunction2, is it possible for your hashMapSize function to return different values?

No, it would not be possible. The hash map size will be constant. The functions just have a different efficiency then the other, but will overall have the same size being used from the same input file.

4. When you run your program on the same input file once with hashFunction1 and once with hashFunction2, is it possible for your hashMapTableLoad function to return different values?

Yes, I believe that the load could be different. This would be based on any or the number of collisions. HasFunction2 is more efficient and in my opinion would have a lower load then the HasFunction1. By having less links in a linked list would allow for more constant time in calculation.

5. When you run your program on the same input file once with hashFunction1 and once with hashFunction2, is it possible for your hashMapEmptyBuckets function to return different values?

Yes, one function will result in ore collisions and thus have more empty buckets then the other. This would make it so that the hashMapEmptyBuckets could return different results.

6. Is there any difference in the number of empty buckets when you change the table size from an even number like 1000 to a prime like 997?

A prime number would have a decreased chance in having collisions versus a composite number. The reason being, the index calculation is based on modulus and a prime number can only have two common factors. This would allow for a more direct indexing.