John Carrabino
November 20th 2016
CS 261 Assignment 6

Hash Table Implementation & Concordance

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

hashFunction1 simply sums the values of all characters passed to it from the input stream. So, passing any two words that contain the same letters to hashFunction1 (i.e. "dog" and "god") would have the same hash. However since hashFunction2 multiplies the values of all of the characters based on that characters index value + 1, it would produce different hashes for two words that contained identical letters.

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

This observation makes hashFunction2 superior to hashFunction1 because it will generate unique values for unique strings, which hashFunction1 fails to do. By being able to generate more unique values hashFunction2 will produce less collisions than hashFunction1, and subsequently produce more buckets with shorter chains.

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?

hashMapSize refers directly to the number of hashLinks in the table. Since each unique key can only be inserted into the table once, regardless of the table's size or which hash function is being used, if each table was passed the same word set they would still produce the same amount of hashLinks.

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?

hashMapTableLoad simply returns the number of links divided by the number of buckets in the current table (which is equivalent to the tables current capacity). We have already established that hashMapSize would not differ between the two hashFunctions, and since capacity is determined when the hash map is initialized and grows dynamically with the structure based on the number of unique entries added to the hash map, the hash functions would also not affect the capacity of the hash map. Since the hash functions do not affect the size or capacity of the hash map then hashMapTableLoad would return the same value regardless of which hash function was used so long as they received the same input file.

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, the number of empty buckets are effected by the hash function being used. In question 1 we talked about how hashFunction2 can create more unique values for words that hashFunction1 would generate the same value for. Since hashFunction2 creates more unique values then its hash map would contain more buckets with fewer chains, and a hash map using hashFunction1 would contain less buckets with more chains.

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?

Since the hash map places new buckets based on the modulo of the key's hash value and the hash map's capacity, there will be less empty buckets for a hash map whose capacity is a prime number. This is because there will be fewer common factors from the modulus operation, resulting in fewer collisions and more buckets being added to the hash map.