Keisha Arnold
Assignment 6
Written

1. Give an example of two words that would hash to the same value using hashFunction1 but would not using hashFunction2.
   The words “saw” and “was” would hash to the same value using hashFunction1, but would not using hashFunction2. This is because hashFunction1 maps the key into an integer, but when they are combined they map to the same key such as with “tea”, “eat”, and “ate”. Hashfunction2 remedies this by also shifting the key so when they are folded words containing the same letters won’t hash to the same key, thus avoiding collisions.

2. Why does the above observation make hashFunction2 superior to hashFunction1?
   One of the main goals of a hash function is that we want it to be fast. Given the above observation, hashFunction2 is superior because it uniformly distributes keys to all indices, meaning there are less collisions and the table size is equal to or slightly larger than the number of elements. This means there are less links per bucket to search through and allows for O(1) time for bag/map operations.

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, the hashMapSize function will return the same value with both hash functions because we still create the same number of links. The hash functions just compute the indices differently.

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?
   On the same note as #3, the hashMapTableLoad won’t return different values for hashFunction1 and hashFunction2 because the hashMapSize and the number of elements will be the same for both hash functions.

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, it is possible for hashMapEmptyBuckets to return different values. As explained in #2, hashFunction2 is superior to hashFunction1 because it uniformly distributes keys to all indices. In other words, hashFunction2 would more likely distribute keys into empty buckets, while hashFunction1 would more likely “cluster” (as shown with “eat”, “tea”, “ate”).

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?
   Yes, a prime number table size would result in a better distribution of indices compared to an even number and we know that a better distribution results in less empty buckets and faster searching.