Jessica Spokoyny
Assignment 6
05/24/2016

# Answers to Written Questions

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

listen, silent

hashFunction1(“listen”) hashes to 5
hashFunction1(“silent”) hashes to 5

hashFunction2(“listen”) hashes to 2
hashFunction2(“silent”) hashes to 9

**2. Why does the above makehashFunction2 superior to hashFunction1?**

With hashFunction2, only palindromes (words which read the same backward or forward) will hash to the same value. Words that use the same letters but in a different order are much less likely to cause collisions and create a better load balance, which leads to faster performance.

For example, using the words: pale, peal, leap, and plea.

hashFunction1(“pale”) hashes to 8
hashFunction1(“peal”) hashes to 8
hashFunction1(“epal”) hashes to 8
hashFunction1(“plea”) hashes to 8

hashFunction1 hashes all 4 words to bucket 8, leaving 9 empty buckets out of 10.

hashFunction2(“pale”) hashes to 4
hashFunction2(“peal”) hashes to 7
hashFunction2(“epal”) hashes to 8
hashFunction2(“plea”) hashes to 9

While hashFunction2 hashes each word to a different bucket leaving only 6 empty buckets out of 10. Clearly the second option is the better one.

**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, hashMapSize returns the number of hashLink in the table. Regardless of which buckets the keys ends up in, the size count will be identical.

**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?**

No, load factor is found by calculating (size/capacity). Regardless of the hashing method, the size and capacity will remain identical. Even if capacity is adjusted for a resize, it will occur with both hashing 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, the number of empty buckets will vary depending on the number of collisions. Using the same example as above:

hashFunction1(“pale”) hashes to 8
hashFunction1(“peal”) hashes to 8
hashFunction1(“epal”) hashes to 8
hashFunction1(“plea”) hashes to 8

Using hashFunction1, hashMapEmptyBuckets = 9

hashFunction2(“pale”) hashes to 4
hashFunction2(“peal”) hashes to 7
hashFunction2(“epal”) hashes to 8
hashFunction2(“plea”) hashes to 9

Using hashFunction2, hashMapEmptyBuckets = 6

**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, because the table size (capacity) is used to calculate the index of the bucket where the key should be assigned. Using a prime number decreases the probability of a collision by eliminating the possibility of common factors of the modulo operator. Thus, in decreasing the number of collisions, we also decrease the number of empty buckets.