Student: Joaquin Saldana

CS261 Assignment 6 Written Answers:

1. **Give an example of two words that would hash to the same value using hashfunction1 but would not using hashfunction2.**

hashFunction1 iterates through the char array, adding the ASCII value of each char until the null character is reached. The problem with this formula is that the words TEA, ATE, and EAT, will sum to the same value, hence the key is not unique to the value.

With hashFunction2, the function takes into account the char's position. It multiplies the char's position with its own ASCII value. In example, in the word TEA, the char T is in the first position so the ASCII value is multiplied by 1. But in the word, ATE, the letter T is in the 2nd position so its ASCII value is multiplied by 2, therefore the sum result of each of these letters is not the same (unique).

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

By taking into account the char's position, we are minimalizing our chances for "collisions" and having the same key for various values, and indirectly improving/maintaining optimum performance O(1).

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?**

Yes, the size of the hashMap utilizing hashFunction1 may be smaller than the size of the hashMap using hashFunction2. Reason for this, is because several words may represent the same key, as a result if the program reads ATE and EAT, both having the same key value, then the program will access that key-value pair and increment it thinking it's the same value.

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?**

Not necessarily. The function finds the ratio (a float number between 0 and 1) between the number of elements (words) in the hashMap and the total number of buckets. The complexity for resizing a list using hashFunction1 and another using hashFunction2 will be the same.

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, since the function returns the number of buckets without any links, the hashFunction1 and hashFunction2 may result in different number of collisions and as a result one hashMap may have fewer buckets than the other.

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 there is a difference. As explained in the slides, initializing the table size to a prime number either lower than the full size of 1000 or slightly higher than the full size of 1000 may result in better load *factors and table resizing occurrences. In finding the hash value, the table capacity is used in the mod to find the hash value and bucket placement. With a prime number, the capacity decreases the chances of a collision occurring since a prime number has only two common factors by definition.*