

Tatyana Vlaskin

Assignment #6

Answer to the questions

1. Give an example of two words that would hash to the same value using `stringHash1()` but would not using `stringHash2()`.

`stringHash1()` sums the values of all of the characters in the word. Thus, any words that have the same letters, regardless of the order of the letters would hash together.

`stringHash2()` sums the values of the characters multiplied by its position in the word. As a result of this, words with the same letters, but in different order will NOT be hashed together, which makes words with the same letters, but different orders unique.

The words “ate” and “eat” will hash together in `stringHash1()`, but not in `stringHash2()`.

2. Why does the above make `stringHash2()` superior to `stringHash1()`?

`stringHash1()` would have more collisions than `stringHash2()`, because in `stringHash1()`, different string that consist of the same letters would hash to the same value. `stringHash2()` resolves this by assigning each character a position, so that there will not be a collision when there are two words with the same letters in different orders. As a result of this in `stringHash2()` more different hashes will be used, leading to a more even distribution and faster performance.

3. When you run your program on the same input file but one run using `stringHash1()` and on the other run using `stringHash2()`. Is it possible for your `size()` function to return different values?

No, the `size()` function returns the count of the `hashMap`. Another way to say it is that the `size()` function keeps track of number of hashlinks added to the table. Regardless of the number of hashes, the `size()` will return same values.

4. When you run your program on the same input file using `stringHash1()` on one run and using `stringHash2()` on another, is it possible for your `tableLoad()` function to return different values?

No, `tableLoad()` is equal to the number of elements in the table (`hashLinks`) divided by its size of the table. Regardless of the hash function, we will get the same # of `hashLinks` and same size of the table. Thus, it is not possible for `tableLoad()` to return different values.

5. When you run your program on the same input file with one run using `stringHash1()` and the other run using `stringHash2()`, is it possible for your `emptyBuckets()` function to return different values?

Yes, different hashFunctions will make words hash to different values. As a result of this, words will be stored in different buckets. This can result in differing amounts of empty buckets.

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, the value a word hashes to is dependent on the table size (it is used with the modulus operator and the returned value of the hash function). If the table size is a prime number, this reduces the possibility of common factors from the modulus operator, which would mean that words hash to the same bucket.

7. Using the timing code provided to you. Run you code on different size hash tables. How does affecting the hash table size change your performance?

I kept getting 0 s until I changed hash size to thousands range. After that, it seems that the larger the table, the slower the process goes because. This makes sense because there are more elements to search through.