

Cole Marquard

cdmarqua@ucsc.edu

- What happens when you vary the size of a hash table?

As you vary the size of the hash table, the lower the size, the longer the linked lists as there are an increased number of collisions that happen due to the small size of the hash table. On the other end of the spectrum, as you increase the size of the hash table, the avg length of the linked lists goes down as with it, the avg search depth also goes down.

- What happens when you vary the Bloom filter size?

As the bloom filter gets larger, there is less of a chance that something might pass through the filter that isn't supposed to. The smaller the bloom filter gets however, the more likely it is to have "collisions"/ words that pass through the filter that are words we don't want to pass through.

- Do you really need the move to front rule?

The need to front rule is not very necessary, however when checking data where words can/do repeat, through the testing of my program I found that the rule helps to lower the avg search depth when looking through the linked lists.

However when there was a large hash table which lowers the amount of collisions/length of the linked lists, the move to front rule is also kind of worthless and barely improves time in those cases. These statics below show that changing

```
hatterspeak@DESKTOP-U06D63R:~/cdmarqua/asgn6$ ./hatterspeak -s -m < oldspeak.txt
Seeks: 14285.000000
Average seek length: 1.746517
Average Linked List Length: 1.898188
Hash table load: 76.709999
Bloom filter load: 0.407801
hatterspeak@DESKTOP-U06D63R:~/cdmarqua/asgn6$ ./hatterspeak -s < oldspeak.txt
Seeks: 14285.000000
Average seek length: 1.746517
Average Linked List Length: 1.898188
Hash table load: 76.709999
Bloom filter load: 0.407801
```