Nout Reusken

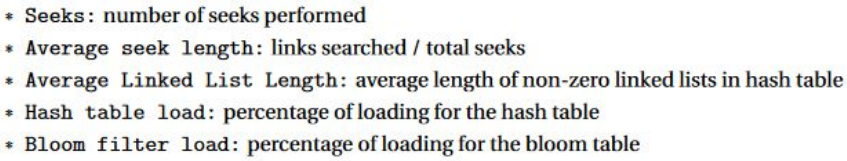
Professor Dunne

CSE 13S

30 November 2020

Assignment 6 Write Up

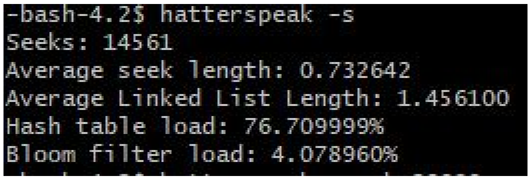
In this assignment, the user can change multiple things, the size of the hash table, the size of the bloom fitler, and whether or not to use move to front rule when looking up in linkedlist. By running hatterspeak -s in the command line it will give you a range of statistics.



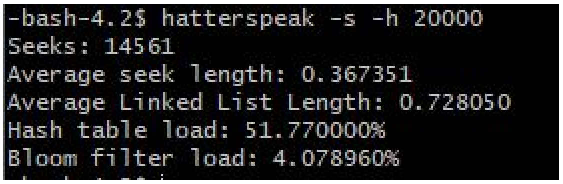
- `Seeks`: number of seeks performed
- `Average seek length`: links searched / total seeks
- `Average Linked List Length`: average length of non-zero linked lists in hash table
- `Hash table load`: percentage of loading for the hash table
- `Bloom filter load`: percentage of loading for the bloom table

This allows us to see how using different sizes of hash tables and bloom filters or different lookup implementations can change the outcome of these statistics.

For the hash table, when looking at the data, with using the default which is 10000 and with a size of 20000, there is a difference.



```
-bash-4.2$ hatterspeak -s
Seeks: 14561
Average seek length: 0.732642
Average Linked List Length: 1.456100
Hash table load: 76.709999%
Bloom filter load: 4.078960%
```



```
-bash-4.2$ hatterspeak -s -h 20000
Seeks: 14561
Average seek length: 0.367351
Average Linked List Length: 0.728050
Hash table load: 51.770000%
Bloom filter load: 4.078960%
```

The data shows the average seek length was half as long when the length of a hash table doubled. This is also seen in the average link list length, when making it twice as long the linke list length decreased half. So this shows the longer, the shorter the seek length and linked list length becomes. But also taking up more space. So if possible try to make it as long as possible but don't make it too big because there migh be memory issues.

Looking at the data when changing the size of the bloom filter, there is not much to see. Since it is just a filter changing the size isn’t necessary since the current size is big enough for this program and a bloom filter does not take up much space so memory is not an issue. So I would recommend just keeping the bloom filter size default.

About necessity the move to front implementation when looking up within a linked list. When usually this would be really beneficial since it would decrease the lookup time when looking up keys in a linked list, for this program it migh not be necessary. Since there linked list in this program aren’t on average very long, 1-2 on average with the default length, so often the key looking for is already in the front. So we don’t really need it for this assignment but recommend it when dealing with long linked lists that have to be accesed often, this would really optimize the code.