Thursday, June 3, 2021

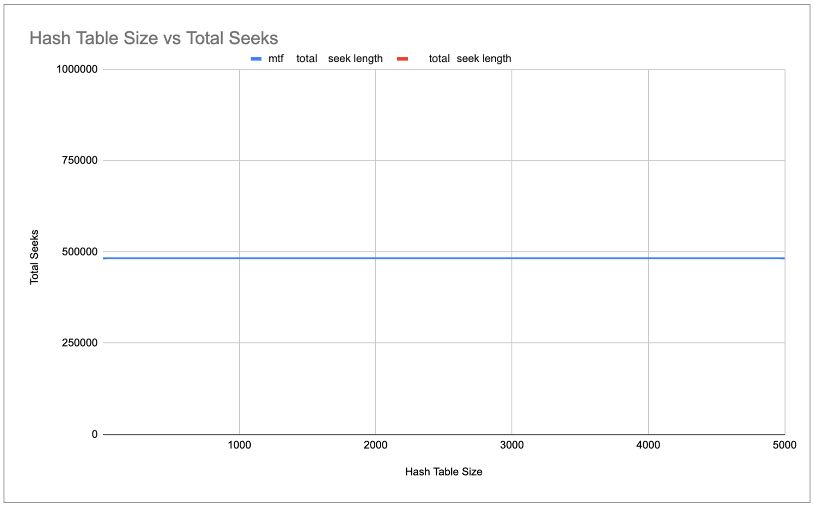
Assignment 7: The Great Firewall of Santa Cruz

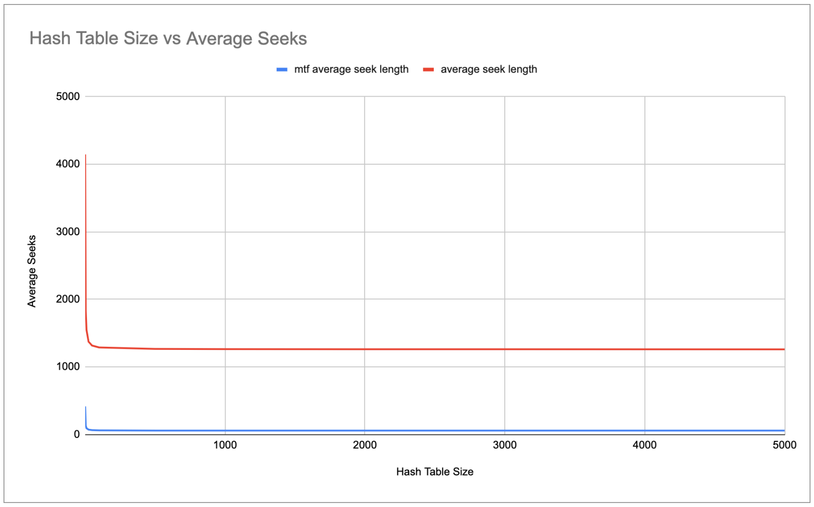
Writeup

- For assignment 7, we wrote a program that filters the users entered words and notifies the user of any words that are not allowed to be used. The main data structures the program uses to filter are hash tables and a bloom filter. They are effective in efficiently searching or items.
- For all of the statistic collecting for the graphs, I used bible.txt as the input text and the given newspeak.txt and badspeak.txt as the filtered words collection.

Hash Table Size:

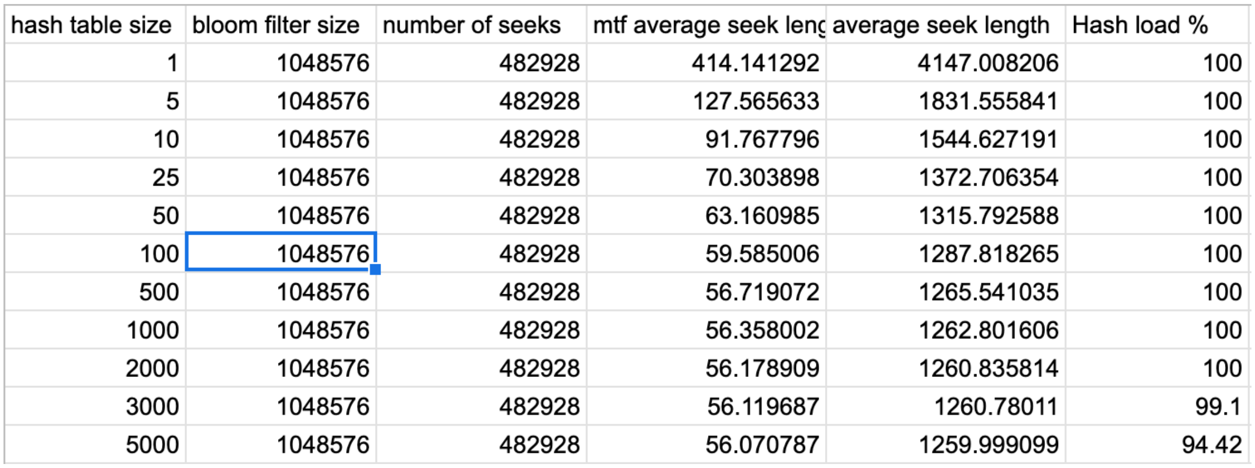
For these graphs, I tested a range of hash table sizes and graphed the total seeks (top) and average number of seeks (bottom).





For the top graph, we can notice that the total number of seeks are identical both when mtf is checked and not checked. The total number of lookups performed in the program has no relation to whether mtf is selected or not. The amount of lookups are also identical no matter the size of the hash table. This is because the amount of lookups performed is dependent on Bloom filter size. In this case, for all tested hash table sizes, we used the same Bloom filter size so then the total seeks should be consistent across all hash table sizes

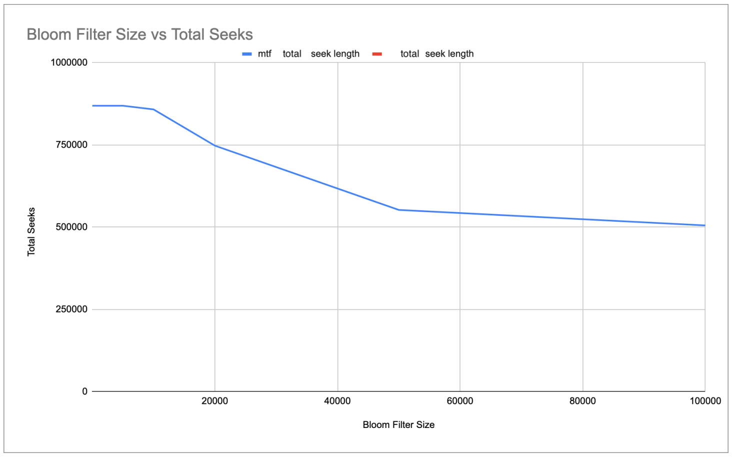
However, as you see from the bottom graph, the number of **average seeks performed per look up is greatly less for when mtf is selected than if mtf is not checked. This is because mtf makes our linked lists have the most recently/ commonly looked up items are the beginning, significantly lessening the number of links to find the word that is being searched.** Both (mtf and not mtf) share a similar pattern when the hash table size increases. As the size of the hash table increases from 1, there is sharp decline in the number of average seeks until the size is around 500+. At this point, mtf average gets very close to a number around 56, non-mtf average gets very close to a number around 1260 if the size of the hash table increases any larger. Both mtf and non-mtf graph hold a shape similar to a negative exponential graph, possibly due to how both will have the same amount of linked lists. The more linked lists there are, the shorter each linked list needs to be, so then less traversing nodes need to happen. Likewise, the smaller our hash table is, the longer each of the linked lists are in the table. More linked lists let us search for the word in a smaller pool of words, but eventually, if there are more linked lists than needed, the lookup seeks won't keep decrease as quickly as when the hash table size was really small.



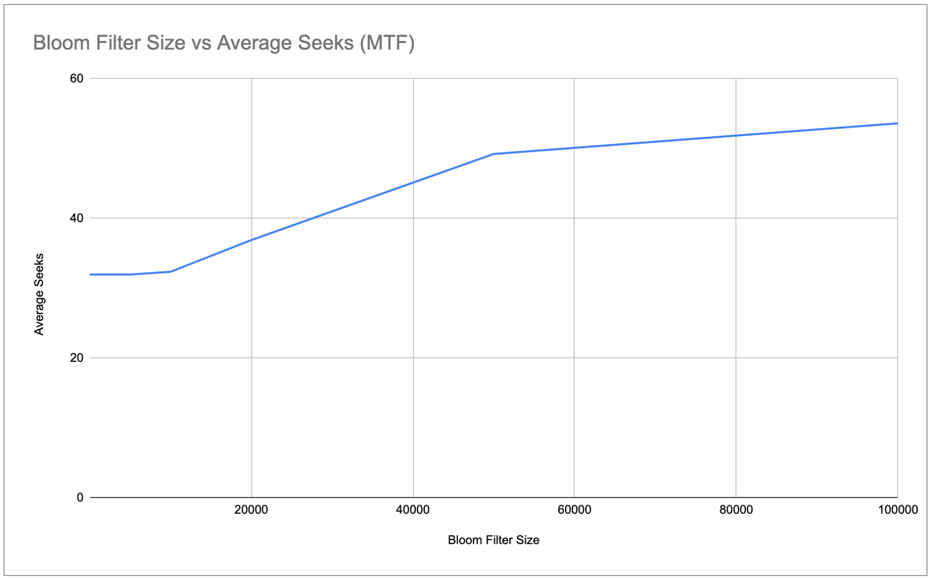
hash table size	bloom filter size	number of seeks	mtf average seek leng	average seek length	Hash load %
1	1048576	482928	414.141292	4147.008206	100
5	1048576	482928	127.565633	1831.555841	100
10	1048576	482928	91.767796	1544.627191	100
25	1048576	482928	70.303898	1372.706354	100
50	1048576	482928	63.160985	1315.792588	100
100	1048576	482928	59.585006	1287.818265	100
500	1048576	482928	56.719072	1265.541035	100
1000	1048576	482928	56.358002	1262.801606	100
2000	1048576	482928	56.178909	1260.835814	100
3000	1048576	482928	56.119687	1260.78011	99.1
5000	1048576	482928	56.070787	1259.999099	94.42

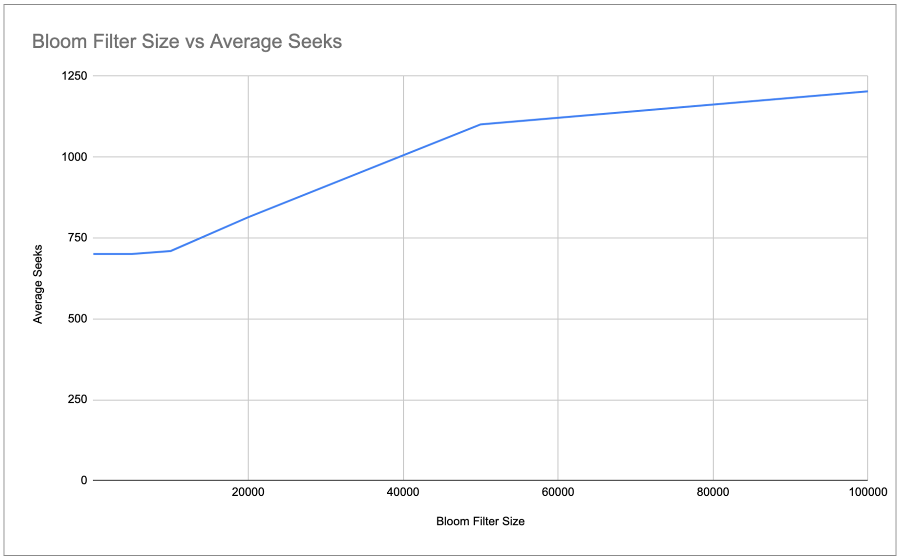
(this is the data table for the entirety of the hash table size graphs since the graphs don’t scale very nicely for the small number)

Bloom Filter Size:

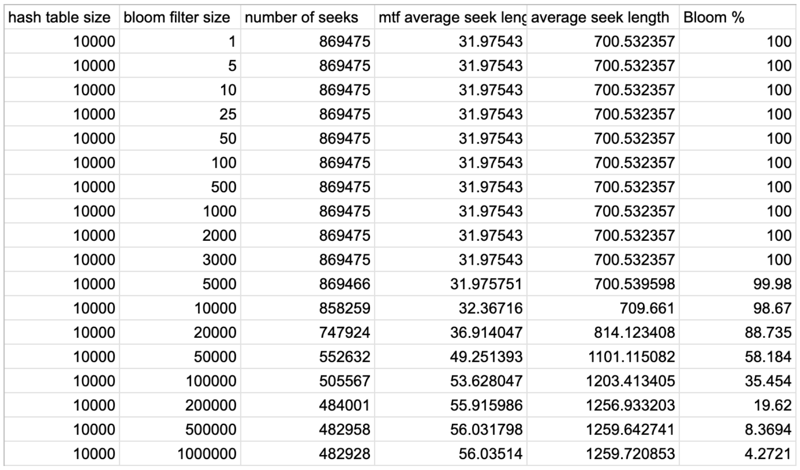


This graph compares the bloom filter size and total number of seeks. Again, the number of lookups between mtf and non-mtf are the same because the number of lookups depend on the bloom filter size, which is consistent between mtf and non-mtf tests. The total seeks in the graph have a downward trend as the bloom filter size increases possibly due to the higher probability for false positives. **The smaller the bloom filter, the more likely for false positives, which means for larger filter sizes, less false positives will be looked up in our hash table. It makes sense that there will be less total number of lookups.**





Similar to hash table, the pattern for average seeks between mtf and non-mtf are the same, however, **the average seeks for mtf is significantly lower than non-mtf. This is caused by the repetition in words being looked up. With mtf, recently search words will be in the front, which means words that are commonly lookuped will need to traverse many less links to find the word.** For both mtf and non-mtf, the average seeks will increase as the bloom filter size increases because less look ups are being made. As we saw from the first bloom filter graph, the number of seeks (lookups) being made decrease as the bloom filter size increases. The average is calculated by: links/seeks, so if seeks is decreasing, then the average length of seeks is increased.



hash table size	bloom filter size	number of seeks	mtf average seek leng	average seek length	Bloom %
10000	1	869475	31.97543	700.532357	100
10000	5	869475	31.97543	700.532357	100
10000	10	869475	31.97543	700.532357	100
10000	25	869475	31.97543	700.532357	100
10000	50	869475	31.97543	700.532357	100
10000	100	869475	31.97543	700.532357	100
10000	500	869475	31.97543	700.532357	100
10000	1000	869475	31.97543	700.532357	100
10000	2000	869475	31.97543	700.532357	100
10000	3000	869475	31.97543	700.532357	100
10000	5000	869466	31.975751	700.539598	99.98
10000	10000	858259	32.36716	709.661	98.67
10000	20000	747924	36.914047	814.123408	88.735
10000	50000	552632	49.251393	1101.115082	58.184
10000	100000	505567	53.628047	1203.413405	35.454
10000	200000	484001	55.915986	1256.933203	19.62
10000	500000	482958	56.031798	1259.642741	8.3694
10000	1000000	482928	56.03514	1259.720853	4.2721

Here is the data table for all bloom filter graphs, with some extra points that would have made the graph disproportionate for the smaller data points. Another interesting thing we can notice is that the number of seeks is constant until the %

bloom filled is less than 100%. When our bloom filter is of big enough size that less than 100% is filled, the number of seeks performed decreases, average seek length for both mtf and non-mtf increases and also roughly hits a constant number, ~56 and ~1260 respectively, as the size of the filter gets biggerwhich is the same for the hash table numbers.