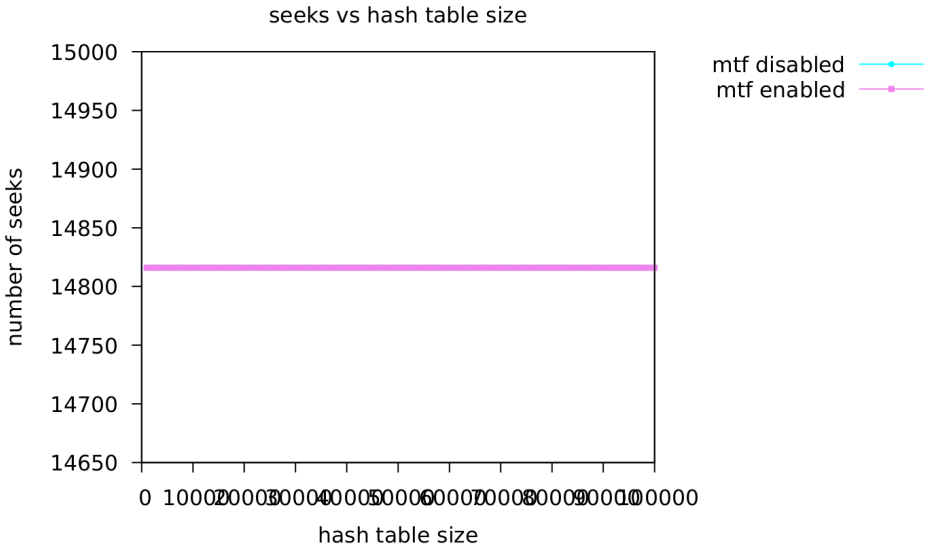
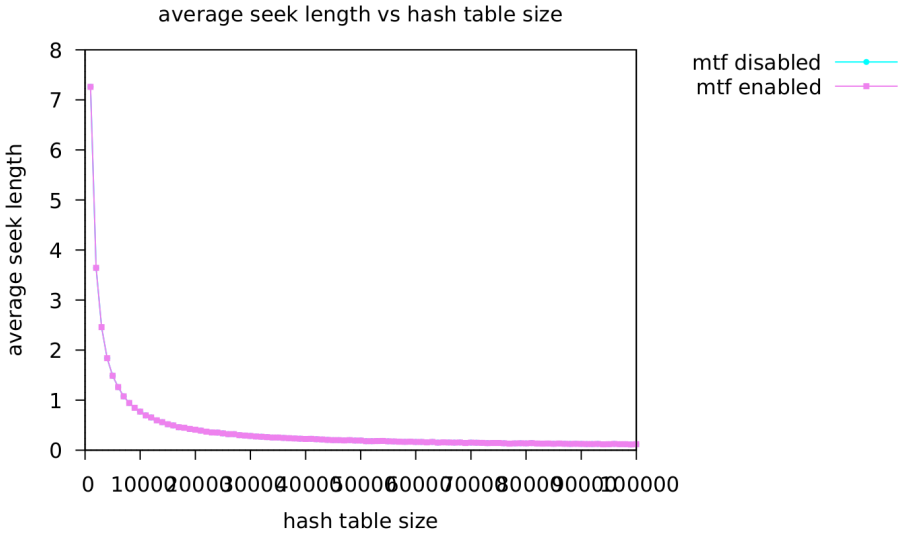
Merilyn Kuo
CSE 13S
Spring 2021

Assignment 7
Writeup

In this assignment, we used bloom filters, hash tables, and linked lists to filter and report bad words. The graphs in this writeup show us a lot of information about the relation between seeks or average seek length and the size of the hash table or bloom filter.

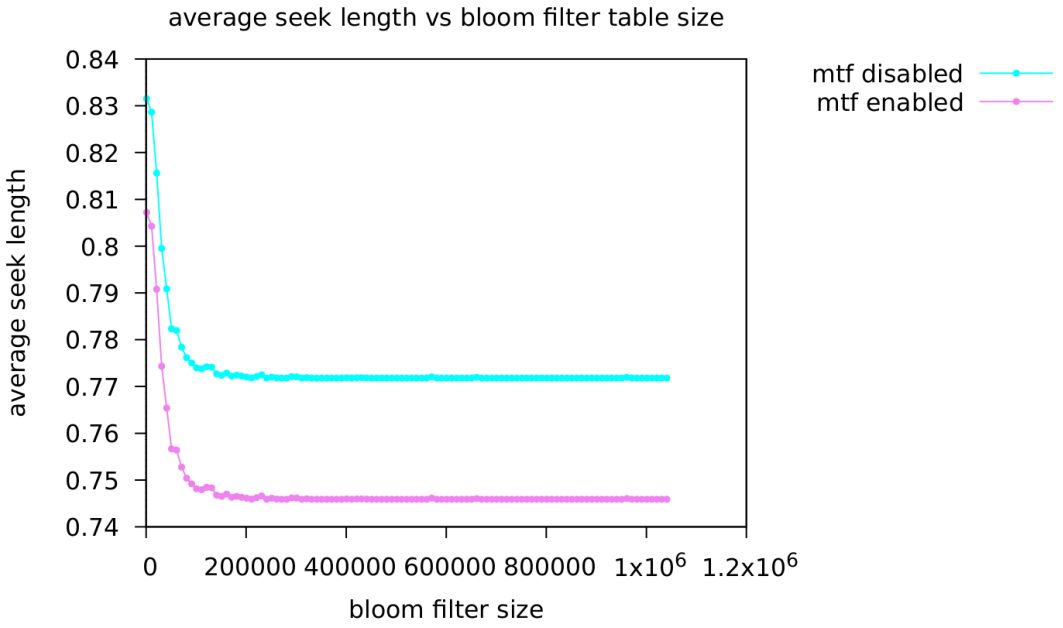


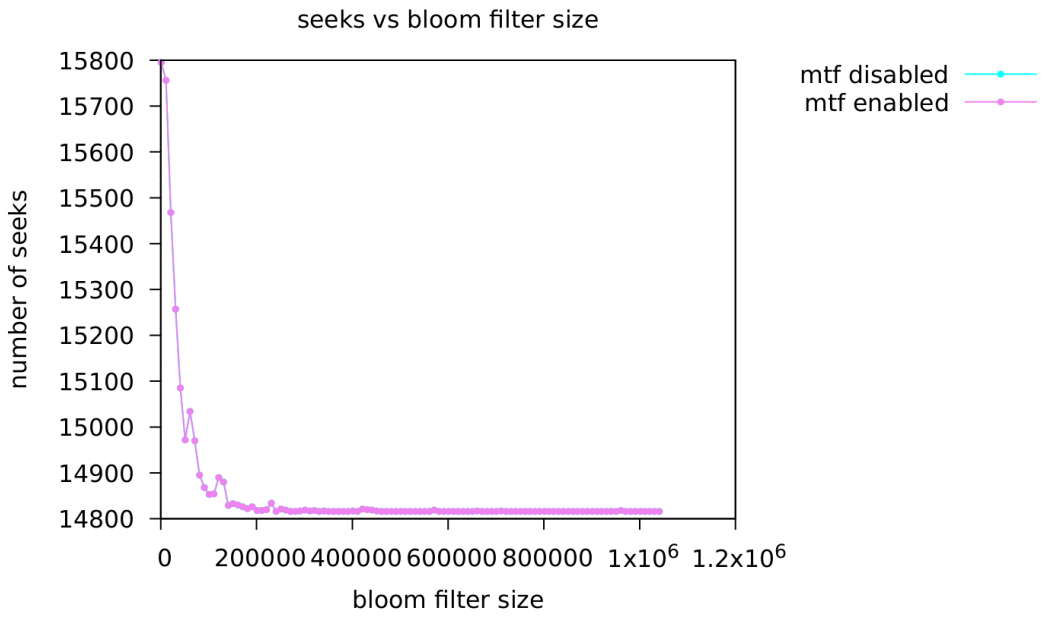


*For the two graphs above, the labeling of the x-axis is a little unclear. The range of the x-axis is 0 to 100,000 in increments of 10,000.

From the first graph, we see that no matter the size of the hash table, the number of seeks the program performs is always the same. Enabling the move-to-front rule also does not affect the number of seeks. I also checked when the hash table size was incredibly small (size of 100) the number of seeks is still the same. The answer to why this happened is simple. The size of the hash table only affects what index is hashed but it does not affect the amount of times a lookup occurs. Once, the index is hashed from the salt and the word, the number of linked list lookups performed either one or zero at that index. And, the amount of times hash table lookups are performed is only affected by the size of the bloom filter. In the two graphs above, the bloom filter size is just the default size so it does not affect the amount of hash table lookups.

In the second graph, comparing average seek lengths and hash table size, we see more of a curve. The average seek lengths were computed by the formula, links over seeks. Since the first graph shows that the number of seeks is always the same no matter the hash table size, the only variables affecting the graph are the number of links and the hash table size. When the hash table size is smaller, the average seek length is larger which also means the linked lists in the hash table are longer. This is because the smaller the hash table, the more likely it is that the hash function mod the hash table size returns the same indices and more nodes are added into the linked list at those indices. As the hash table size grows, they hash into the same indices less. In the second graph, enabling the move-front-rule looks like it does not affect the results as well.





The two graphs above show the effects of the bloom filter size on the seeks and the average seek length. In the first graph, we can see variations due to the size this time. The same information is also shown in the second. This is expected as the smaller the bloom filter, the more likely it is for the hashes to hash into the same indices and false positives to occur. False positives will result in more hash table lookups than needed and therefore more seeks. The first graph also shows that enabling the move-to-front rule decreases the overall seek length. This can be explained by the fact that the number of nodes needed to check when looking up a word (and especially duplicate words) is less if each time a word is looked up, the word is moved to the front. In short, the move-to-front rule lessens the number of links and therefore the average seek length.