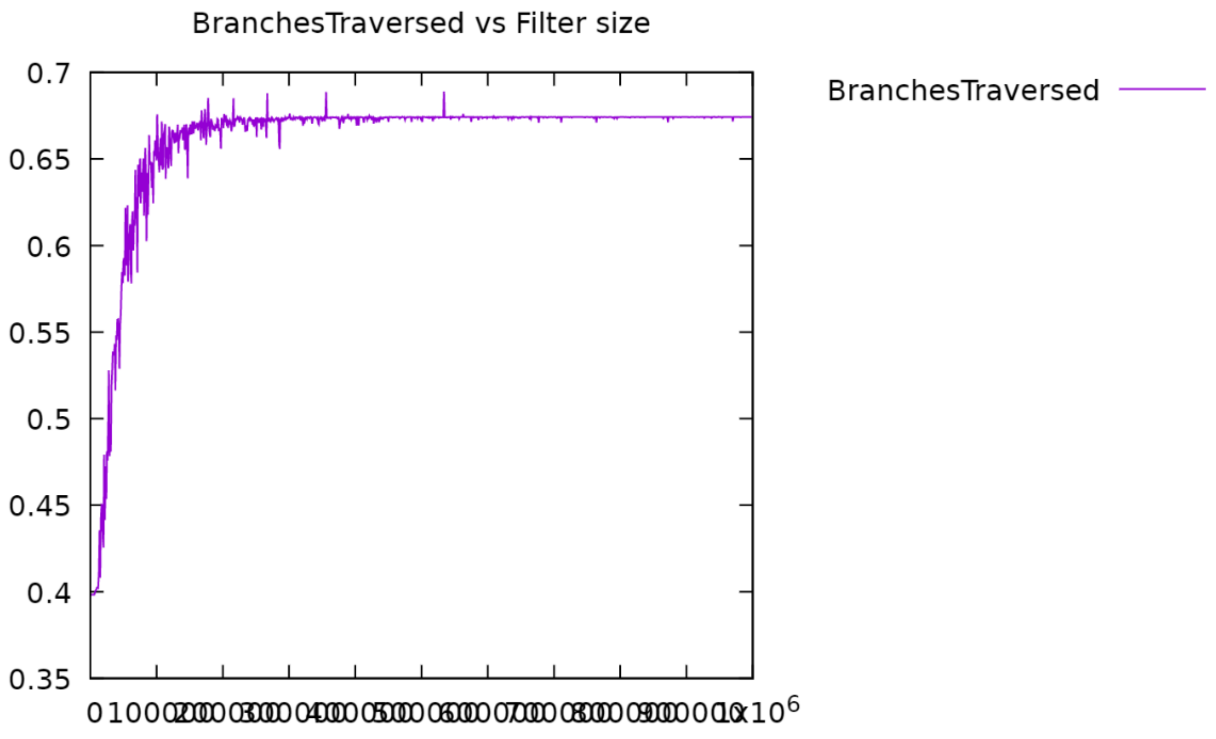
Writeup ASGN 7

Note: This was ran with the alice29.txt

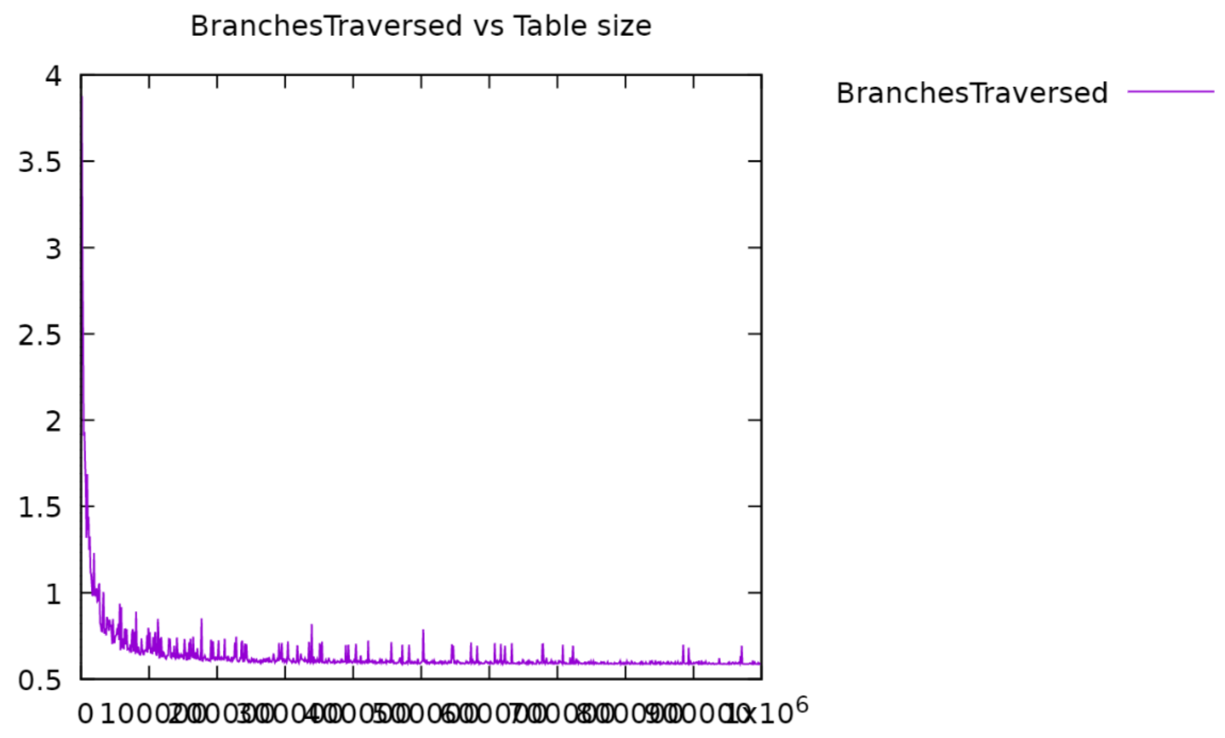
Branches Traversed VS Filter Size



Note: Each tick on the Xaxis increases by 100000

Above is a graph showing the percentage of branches traversed as the filter size increases from **1000 to 1000000 at a rate of 1000 uint32 integers per iteration**. As seen from the graph above, the number of branches traversed exponentially increases from approximately 1000 to 150000. During that time, the number of branches traversed react sperastically and the trend is unstable. As the filter size increases however, the trend begins to stabilize and hits a mathematical asymptote at around **.68** branches traversed.

Branches Traversed VS Table Size

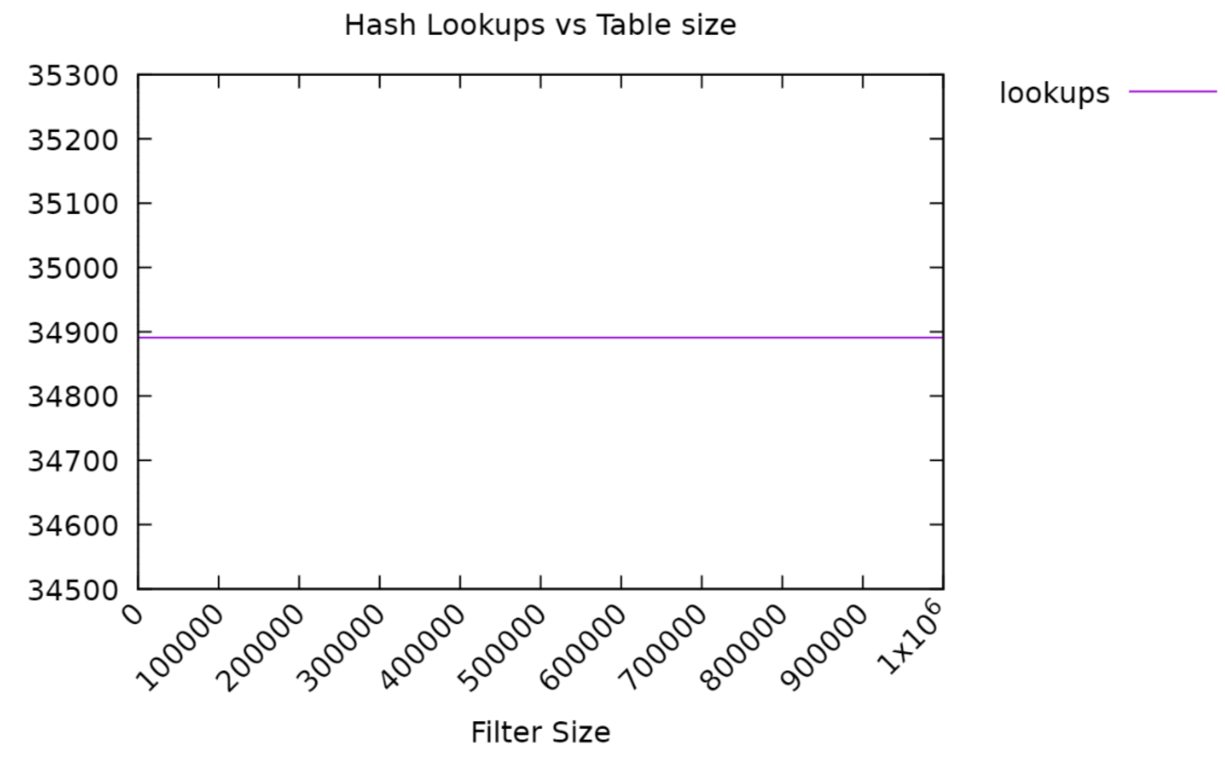


Note: Each tick on the Xaxis increases by 100000

Above is a graph showing the percentage of branches traversed as the table size increases from **1000 to 1000000 at a rate of 1000 uint32 integers per iteration**. As seen from the graph above, the number of branches traversed exponentially decreases which is the exact opposite of the previous graph. While the graph moves in the opposite direction, it also shows the same stability as the previous graph as the table size increases.

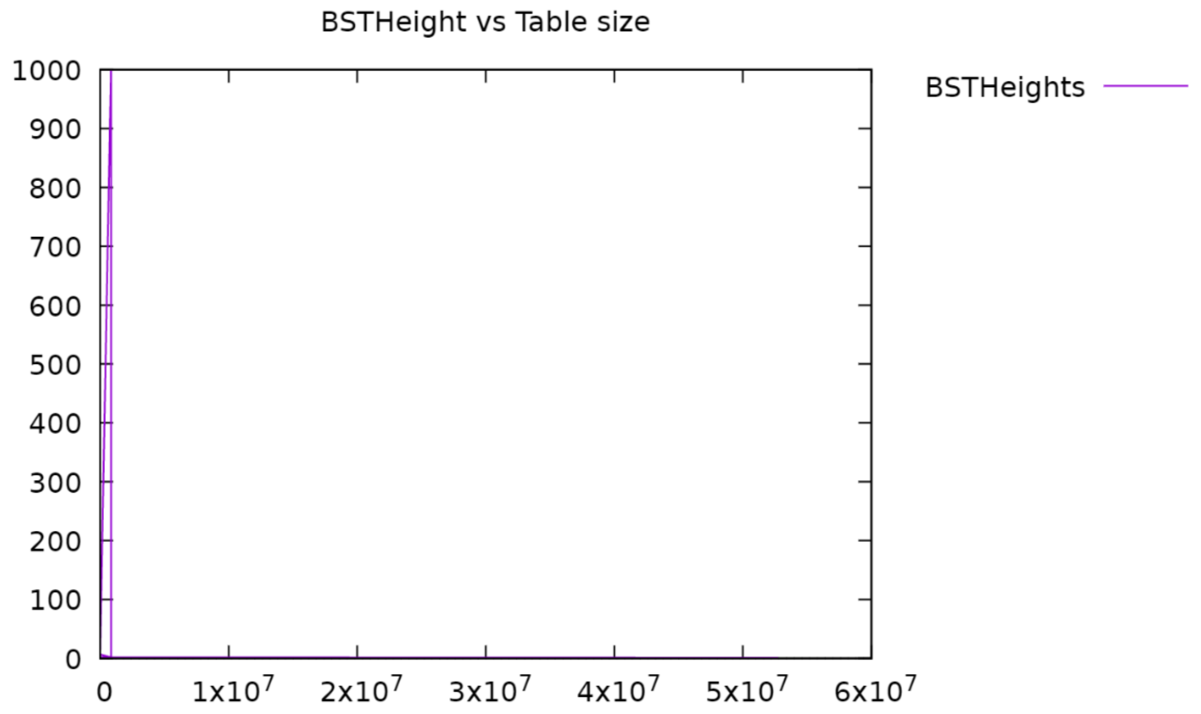
Conclusion:
Both graphs show that if one were to increase the table size and filter size have an inverse relationship to how many branches are traversed in banhammer. This shows the behavior of their original mathematical form which is **branches traversed = filter size/table size.**

Lookups VS Bloom Filter Size



Above is a graph showing the number of lookups in the hashtable performed as the filter size increases from **1000 to 1000000 at a rate of 1000 uint32 integers per iteration**. As seen from the graph above, the number of lookups performs no change as the filter size increases. Therefore the filter size has no effect on the number of hash table lookups implying that as you hash the words, they always go to the same index in the hash table and bloom filter which means the number of hashtable lookups does not change.

BSTHeight VS Table Size



Above is a graph showing the average BSTHeight performed as the table size increases from **1x10^7 to 6x10_7 at a rate of 1000 uint32 integers per iteration**. As seen from the graph above, the average height of the BST dramatically increases at the beginning then drops as the table size increases. This implies that as the height of the hashtable increases, there is more space for words to be stored as more and more trees are added. As a result, the average tree has to store less and less data which means that their height decreases.