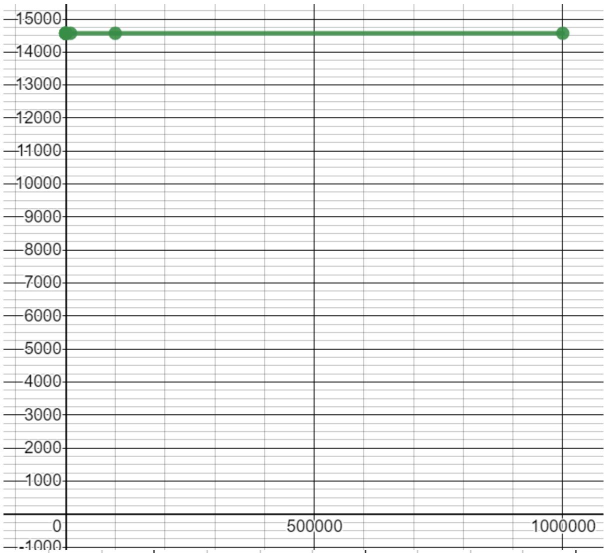
ASGN7: The Great Firewall of Santa Cruz:
Bloom Filters, Linked Lists, Binary Trees and Hash Tables

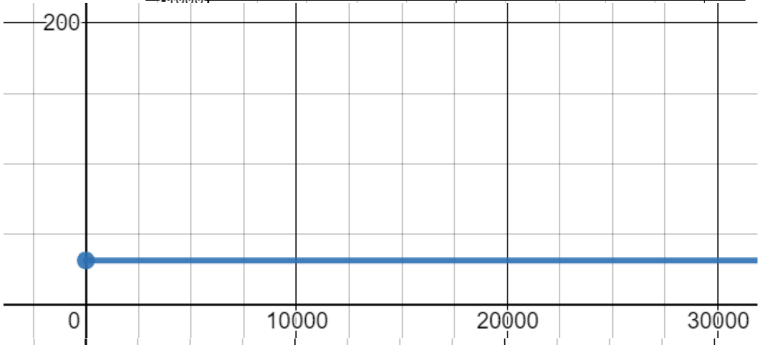
The main two elements of banhammer.c that comprised most of the logic behind calculating whether a word was bad were the hashtable and bloom filter adt. The default size for the hashtable was 65,536 and the default size for the bloom filter was 1,048,576. But why include the ability for the user to change the size of either adt? The reason for that can be seen and realized after analyzing several graphs.

If we alter the size of just the bloom filter we can see the effects on the average bst size, average bst height and average branches traversed statistics with these graphs.

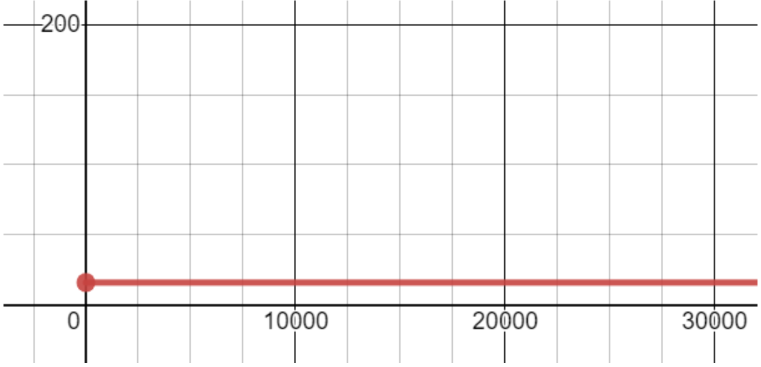
BST size:
X-axis: Size of BF



BST height:



BST branches
Traversed:

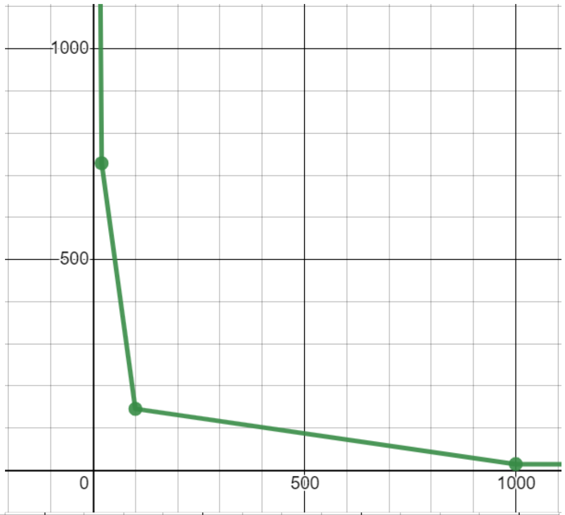


As can be seen by all three graphs the three statistics relating to the bst are unchanged regardless of the size of the bloom filter. The reason for this is due to the sole purpose of the bloom filter, it does not store any of the bad words and their translations. It only stores a hashed value of the bad word, this means that the only purpose of having a larger bloom filter is to decrease the chances of two bad words sharing the exact same position in the bloom filter. That in turn decreases the chances of a user inputted word being incorrectly flagged as a bad word, saving the computer the time of needing to search the hash table for the bad word and it's translation.

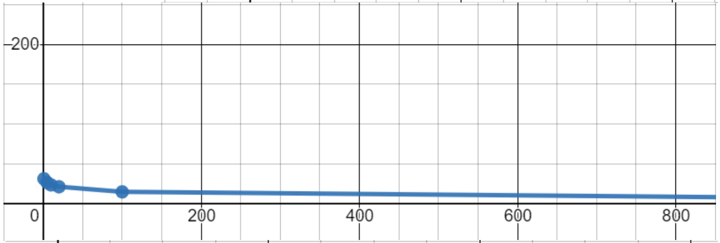
The fact that the size of the bloom filter does not change the value of any outputted statistics of course means that only the hashtable's size can change the statistics. Those statistics are as follows-

BST size:

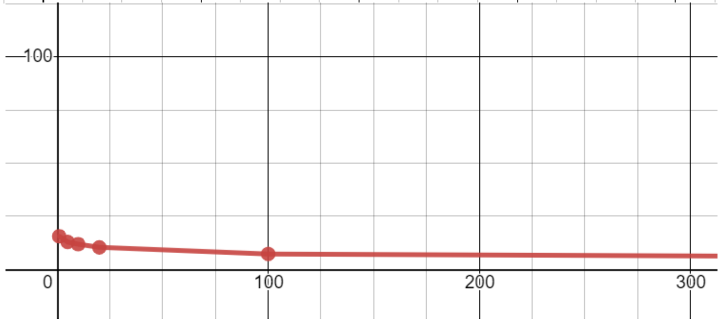
X-axis: size of hash table



BST height:



BST branches Traversed:



As evidenced by the previous graphs, a larger hash table results in a smaller average bst size, height and branches traversed. With smaller values, the time needed to find (or not find in the case of a false positive) a word in the hash table decreases with larger and larger hash tables.

Increasing the size of both the bloom filter and the hashtable ultimately increases the speed at which banhammer can check all of the user's inputted words. Increasing bloom filter's size decreases the odds of a false positive, reducing the number of unnecessary probes to the hashtable. Increasing hashtable's size increases the chances of each bad word receiving its own binary search tree which decreases the overall height of every bst, reducing the amount of time needed to search through a single bst to find the requested word.