Dmitrius Agoston
Prof. Darrell Long
5 December 2021

CSE 13S Fall 2021
Assignment 7: The Great Wall of Santa Cruz
Writeup

Abstract:

The objective of this assignment was to have a method to filter messages and potentially punish anyone who is caught by the filter. The methodology used was creating a bloom filter and corresponding hash table to filter the given messages. These data structures used were possible through the use of binary search trees. These binary search trees statistics were recorded and then analyzed by varying the size of both the bloom filter and hash table.

Introduction:

The purpose of this assignment was to investigate the methods behind filtering messages through the use of a bloom filter and hash table. These two data structures both rely on the use of binary search trees in order to function. The purpose of these trees is to give a quick and efficient data structure that can be used to store and find the words we want to filter with the translation to replace the word with if applicable. To fully investigate these binary search trees the statistics of their data was recorded and analyzed. This was done through gathering the values of various elements of the binary search trees like the average branches traversed, lookups, and height of the trees with various sizes for both the bloom filter and hash table. The results of this investigation show that a change in bloom filter mainly only affects the number of lookups whereas a change in hash table mainly affects both branches traversed and height.

Materials and Methods:

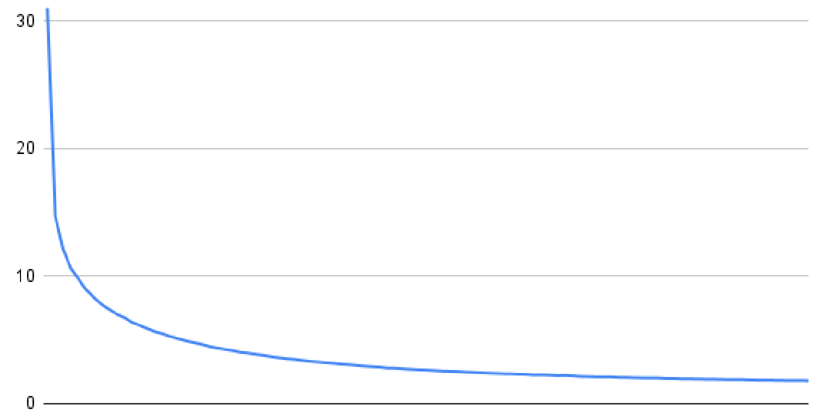
The main methods used to test these sorts were implemented in the test harness that was created which was aptly named banhammer. The test harness was responsible for deciding what size to make the bloom filter, what size to make the hash table, and to give the statistics at the end or not. The test harness was run about 100 times for each statistic, except for average lookups with a changing bloom filter, which needed to be run about 1000 times for the given results.

Results:

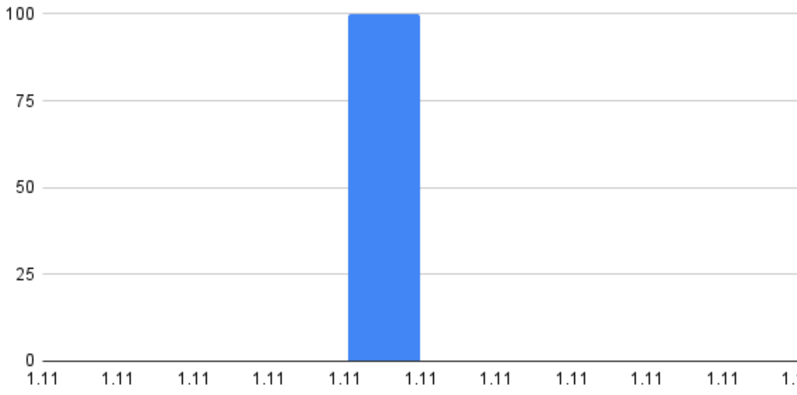
After analyzing the data given from the tests I found that there was one main factor that affected the average height of the binary search trees. This factor was dependent on whatever the size of the hash table was set to. When the hash table was given a size of

1 the average height of the binary search trees was 32 whereas a hash table with the size of 10,000 had an average size of 1.8. When analyzing the graph it is clear that the average height sharply declines when the size of the height table increases. Also after a certain point the average height changes becomes pretty negligible after the hash table reaches a size of 10,000. There was also the data when the bloom filter's size was changed. Though all the data received from every 100 variations in size of the bloom filter was exactly the same, with an average size of 1.110016. Overall this leads me to believe that the leading factor for binary search tree height is due to the size of the hash table given.

Average bst Height with Changing Hash Table (1-10,000)

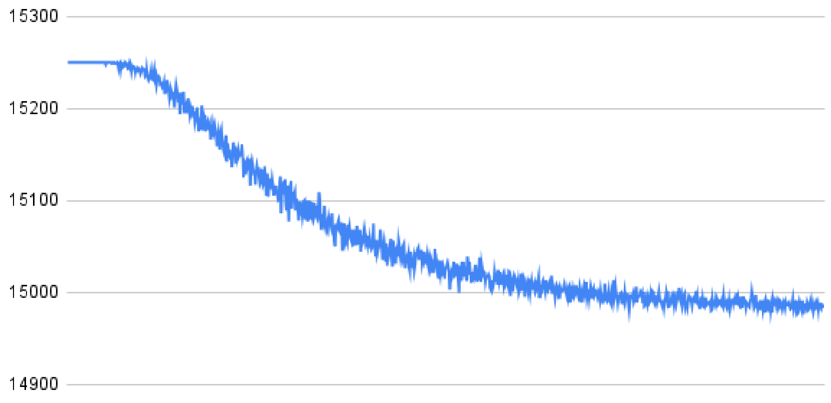


Histogram of Average bst Height with Changing Bloom Filter (1-10,000)

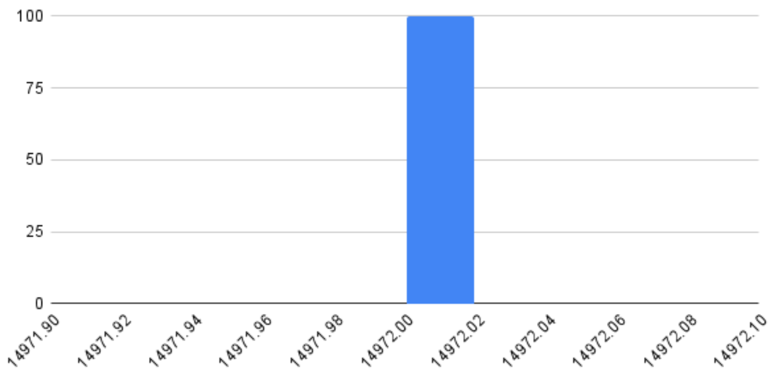


When analyzing the statistics given from changing the bloom filter there seemed to be only one data set that was noticeably affected. This was the average number of lookups that had occurred on the binary search trees. When first analyzing the graph it seems that about the first 50 increases to bloom filter are not significant enough to change the average number of lookups. Though after these 50 increases the graph begins to form a slope and the average number of lookups trend downward. Towards the end of the test it would also seem that this decrease in lookups gets less and less significant after the bloom filter is set to around 100,000. The average number of lookups was also

Average bst Lookups with Changing Bloom Filter (1-100,000)

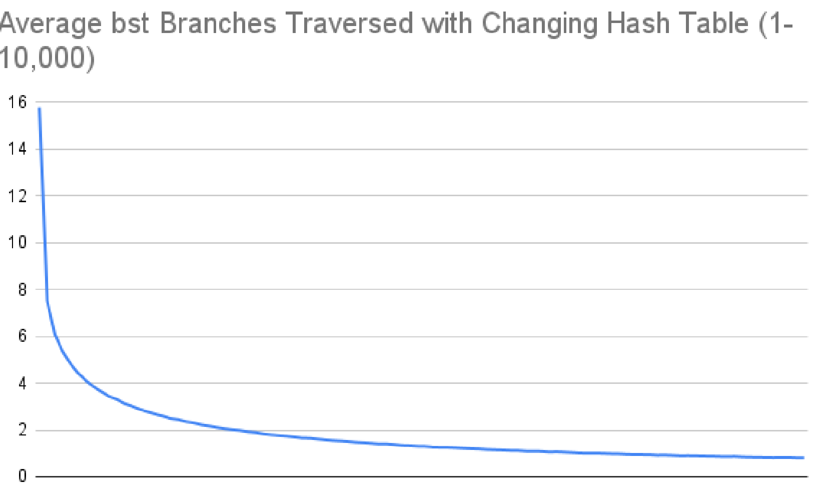


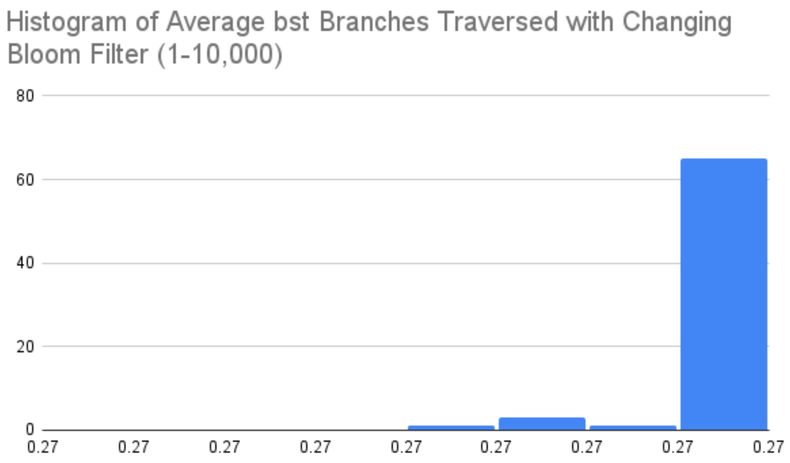
Histogram of Average bst Lookups with Changing Hash Table (1-10,000)



compared to the hash tables size changing though these results were all exactly the same with no variation. The lookups consistently sat at 14,972 lookups on average when the hash table size was changed.

The hash table also seemed to be another major factor for one of the elements of the binary search trees. This element in particular is the average number of branches that were traversed. When given a hash table with a low value the average number of branches traversed falls around 15. Though when the hash table's size is increased, the average number of branches traversed decreases drastically. Much like many of the other tests, after increasing the size of the hash table to 10,000 the differences in averages were pretty negligible. Interestingly enough though when the bloom filter was altered, the differences in average were negligible all across the board. Though looking at the graph, there was not a constant number and there was at least some slight variation in the number of average branch traversals.





Conclusion:
What I can conclude from these tests and the data gathered is that, when looking at the elements of binary search trees analyzed, it seems that either the size of the bloom filter or the hash table affect the average, not both. Looking at the opposing data structure size change either doesn't affect the change in average at all or so little that the average difference is negligible.