Eduardo Garcia
egarc127@ucsc.edu
12/05/2021

CSE 13S Fall 2021
Assignment 6: The Great Firewall of Santa Cruz:
Bloom Filters, Binary Trees and Hash Tables

Write Up Document

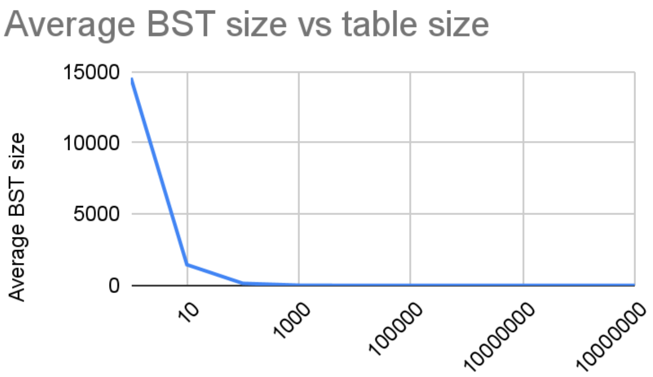
Introduction

In this document we will explore how different size for our Hash Table and Bloom filter affect the size and heights of our binary search trees, Hash table and Bloom Filter load, and average branches traversed.

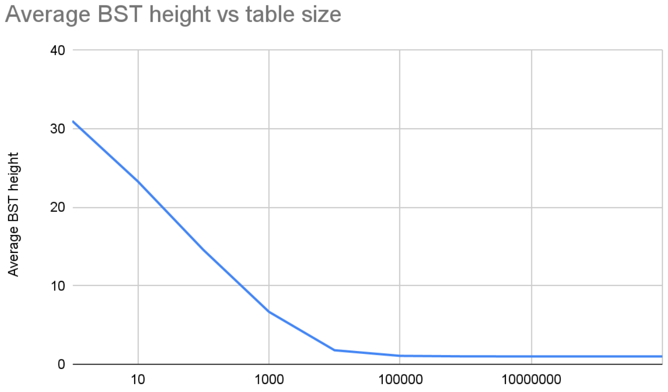
Definitions and control variables

All tests will be performed using the provided newspeak.txt and badspeak.txt
The standard Hash Table size is 2^16(65536)
The standard Bloom Filter size if 2^20(1048576)
BST size is how many non-null nodes in a Binary Search Tree
BST height is the longest chain of nodes pointing to other nodes
Table load is a percentage of ht_count/ht_size
Filter load is a percentage of bf_count/bf_size
All tables will be shown using a logarithmic x axis
Test are done between 10^0 and 10^9

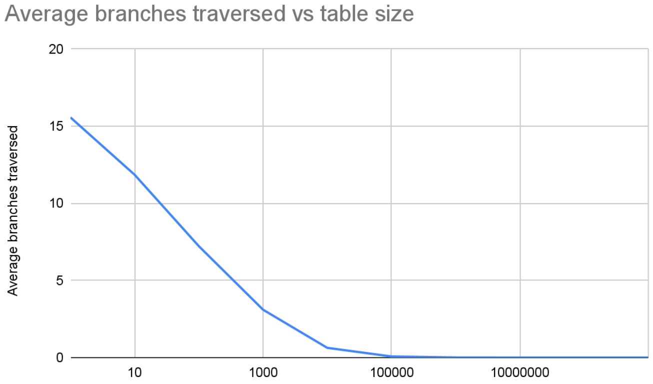
Changing the Table Size



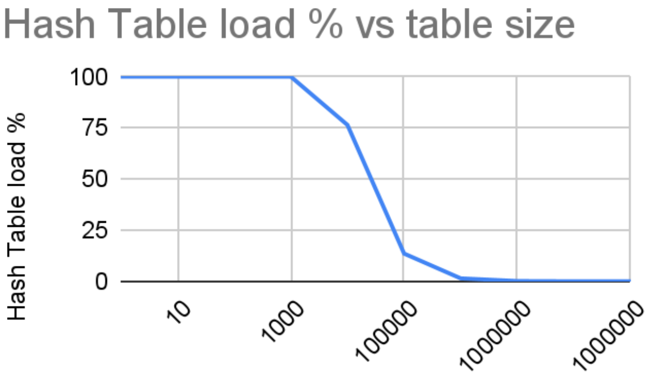
As we can see, the BST size goes near 0 once we have a table of 100, and after that it doesn't matter how much bigger it becomes.


Average BST height vs table size

The average height of our BSTs drops off gradually, and at 10,000 is near 0
We can see that a table size < 1000 will have average BST height.


Average branches traversed vs table size

At around 10,00 the amount of branch traversals falls off drastically, this would be around the ideal table size so that we do not have so many hash collisions.

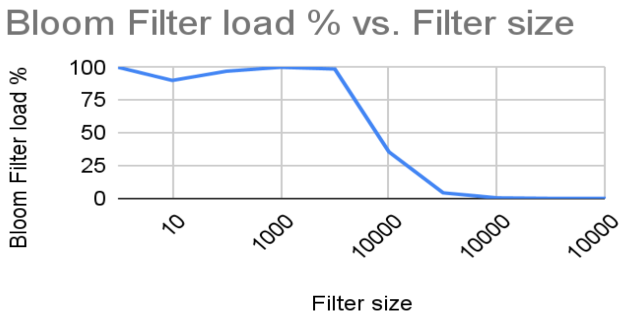

Hash Table load % vs table size

At around 100,000 is when our table load falls to near 0, and at 10,000 falls below 25%. To have a reduced amount of hash collisions, this would be an ideal range without having too large of a table.

Conclusion:

Between 100,000 and 1,00,000 is the ideal hash table size where we have a healthy mix of not too large of a Hash table that space is wasted, and not too large that hash collisions are extremely common.

Changing the filter size



Anything non-100 filter load below the 10,000 Filter size limit should be discarded as this part did not pass the pipeline. After accounting for that, we can see that our filter load is about 5% at 1,000,000 and less than 0.5% at 10,000,000.

Conclusion

At within the range of 1,000,000 and 10,000,000 is where we our filter size to be at. Since this is a bit vector, this number/8 bytes are used, making our programs as space efficient as possible while reducing the amount of false positives.