Assignment 5 Writeup - Jack Vento

Hash Table

The size of the hash table has a direct impact on the efficiency of it. The larger the hash table, the less collisions there will be: leading to smaller linked lists (i.e less chaining) and consequently shorter search times. If the length of a hash table is too small, its performance will quickly degrade as it is directly proportional to the hash table's load factor.

Bloom Filter

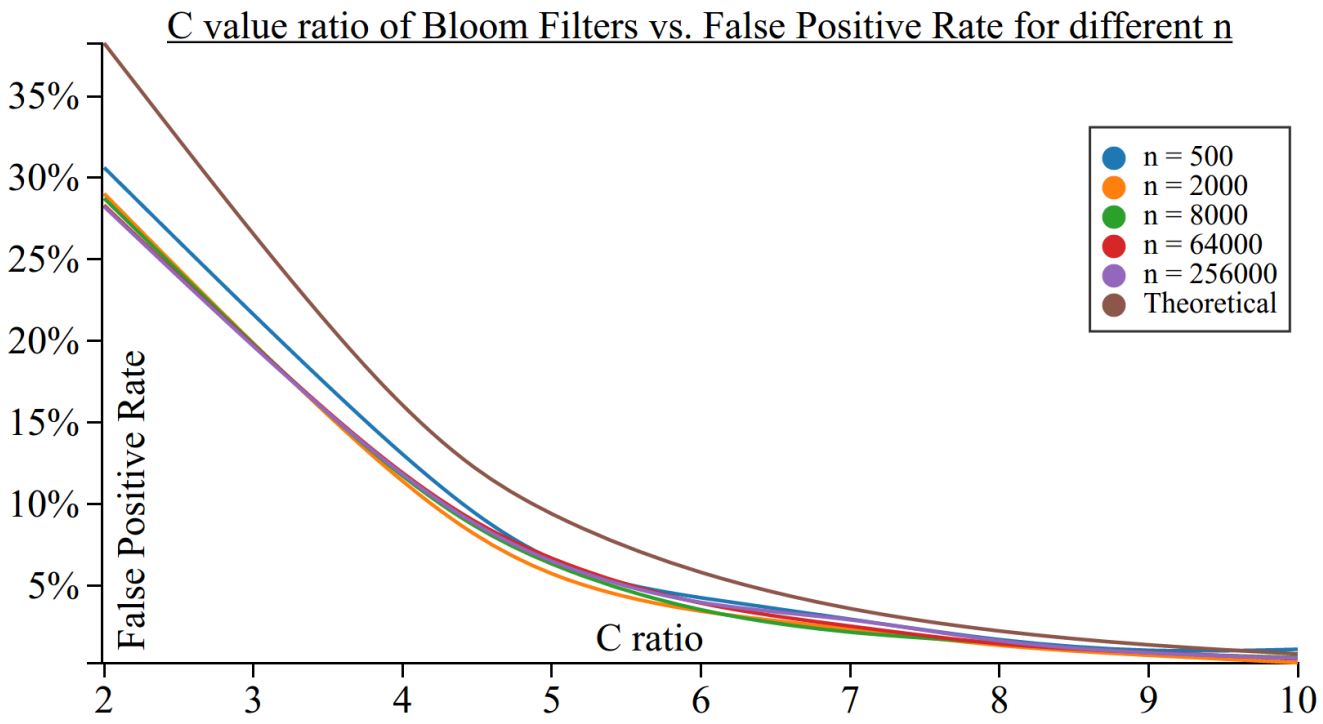
Varying the length of the bloom filter has a direct effect on the probability of the bloom filter probing a false positive. The larger the bloom filter, the lower the probability that a probe returns a false positive; probablility and length are inverses in a BF. However, raising the length does does decrease space efficiency as it requires a larger bit vector filter.

The equation for the probability of getting a false positive from a bloom filter is as follows:

$$P = \left(1 - e^{-\frac{kn}{m}}\right)^k$$

... where n = inserted elements, m = length of BF, k = amount of hash functions.

See the following graph for an illustration of this phenomenon (source):



Move-To-Front

The basis behind the move-to-front technique is taking advantage of temporal locality, meaning that recently used items are likely to be used again in the near future. Theoretically, by moving nodes to the front of the list whenever they are looked up, the most frequently searched nodes will be near the front of the list: leading to quicker lookups.

However, in this assignment, the performance benefit is neglibile as the average length of each non-empty linked list is less than 2 items due to the nature of hashing. Were the assignment using linked lists outside of a hash table, the technique's performance improvement would be much noteable and cache friendly.