Din Bostandzic
CSE 13S

Assignment 6 - Bloom Filters, Hashing, and the Red Queen's Decrees Writeup

The results obtained in this lab were from the implementation of a hash table using a bloom filter and linked list structure, the latter which was used to resolve any collisions in the hash table. A default size of 10000 was used for the hashtable, as for the bloom filter a default size of 2^20 or 1048576 was used. When the program is run with just these default values, the stats are as follows: Seeks: 14563 (the total lookups of a node), average seek length: 0.732267 (total number of links traversed until node found / seeks), Average Linked List Length: 1.455700 (total number of non NULL lists in the hashtable / size of hashtable), Hash table load: 76.710000% (#total filled slots in hashtable / hashtable size), and finally Bloom filter load: 4.166508% (#total bloom filter slots in bloom filter / bloomfilter size).

These results from defaults indicate that the link lists used proved to be efficient as the collisions that occurred were handled and resulted in the hashtable only being about ¾ full. When the hashtable size is lowered to say 1000, it becomes fairly inefficient for storage as the entire table gets filled with hash load of 100%, the average seek length increases by 10x, due to the size being 10x smaller. As the hash table is increased, the hash load is lowered, indicated that hashtables work well for storage when they are bigger because the slots easily getting taken up with lots of data coming in, resulting in longer chains of linked lists, and relying on them. For instance, when hashsize is 100000, 10x bigger than the default, the hashload becomes: 13.5% as the average seek length reduces by 10x. On the other hand when the bloomfilter size is reduced to test value of 104857, the bloom filters load increased like the hashtable behaves, due there being less slots to allow for bits being turned on, being less spread out. The bloom filter load was 41.6%, which is 10x bigger than the default load. When the bloom filter size is increased, to 1000000, which is 10x the size of the default, the bloom filter load goes down to just 0.43%, which is 10x less the default bloom filter load. This is due to the more slots allowing more spread out bits getting set, based on the number of inputs. Finally, the move_to_front rule for moving the searched node to the front of the list is not really needed, as the list will need to be traversed regardless up until that point where the node is located in the list, and only then does it get placed to the front. So using this rule there is no difference as with when it is not activated as the results from the stats output appear to be same.