

## Hash Table Analysis

For my Hash Table, I implemented a hash function that multiplies the key by a sufficiently large number that provides for a more uniform distribution of keys than other functions tried. Since it uses multiplication, it may be considerably slower than other alternatives, but the reduction in collisions compared to the other implementations attempted should more than offset for this. The hash table was increased to a capacity of 1009 records, because a prime number size meshed better with my hash function. Even with this hash function, this hash table still has a seemingly poor performance when the hash table gets more than 60% full. After this point, the number of collisions per record insert grows increasingly exponentially, much faster than the theoretical best performance (75 collisions at 70% up to 260 collisions at 90%). Despite this, for loading factors below 60% full, my hash function manages to stay exceptionally close to the theoretical best. When records were inserted sequentially into the table, no collisions were encountered for the first 400 records. Random insertion performed expectedly worse, but the maximum number of collisions for 400 records was only 19. As this table uses pseudo-random probing to resolve collisions, it suffers from secondary clustering, which was seen to affect the number of collisions in the table after filling to 50% capacity. Figure 1 shows the data collected for both a sequential record insert, a randomized record insert, and the theoretical best performance. Figure 2 removes data for 99% table capacity, to better capture performance at lower capacities. Measured CPU Time of this hash table was recorded against random record insertion (to reflect the potential worst case), and mostly follows expected results based on the number of collisions, within some slight variations (i.e insertions at 99% took less time than insertions at 90%, despite exceptionally fewer numbers of collisions experienced. There is also a drop in CPU Time from 400-500).

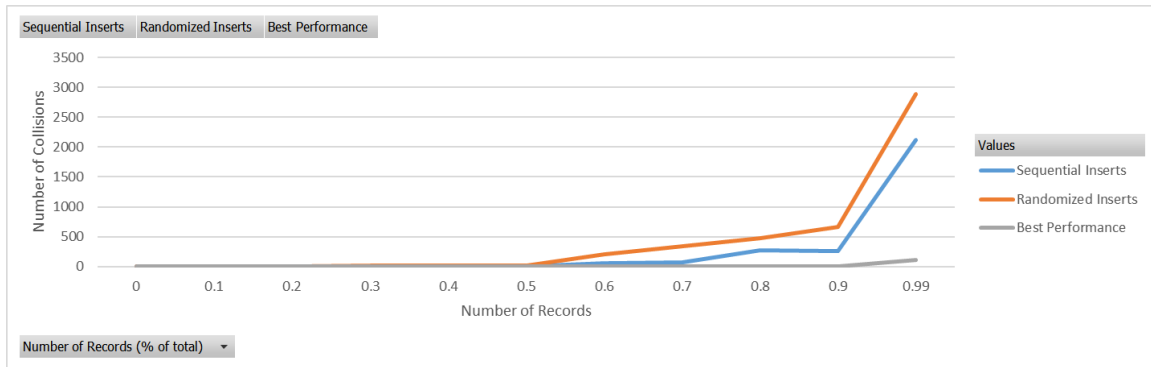


Figure 1 Graph of number of collisions on insert based on hash table capacity.

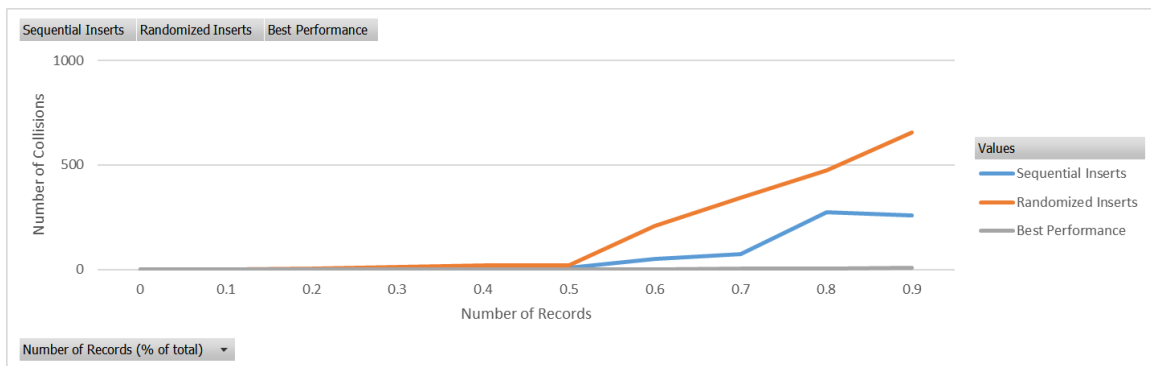


Figure 2 Graph of number of collisions on insert based on hash table capacity (99% capacity removed)

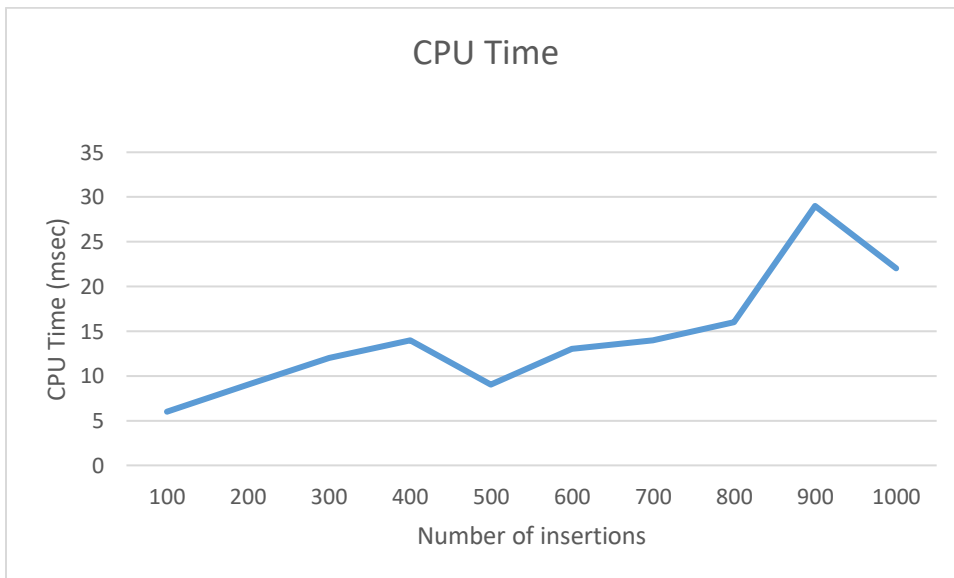


Figure 3 CPU Time vs number of record insertions