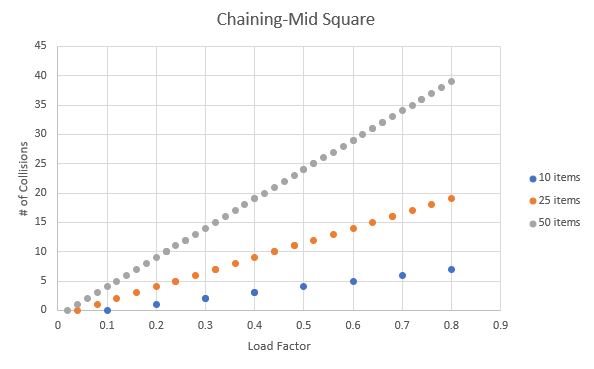
Michael Nunn

Cop3530 | 10/28/2018

Hashing Analysis

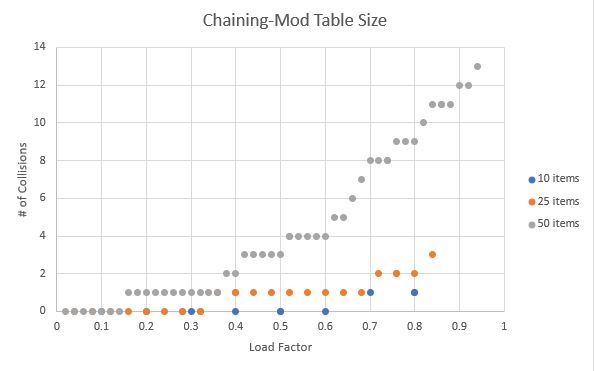
Data Structures and Algorithms

Chaining resolution with Mid Square function



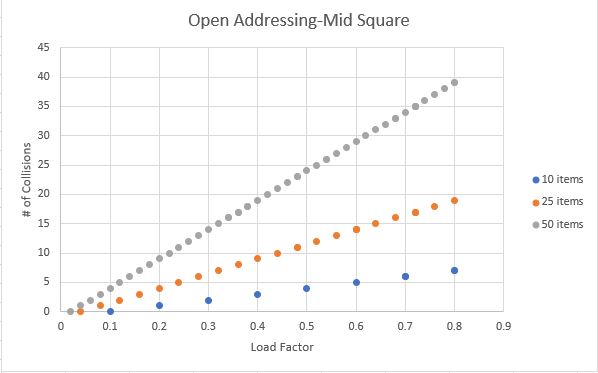
This plot represents a hash table created using chaining with a mid-square key generation function. The load factor is on the x-axis, this represents the quotient of filled buckets/table size. Collisions happen when you attempt to add an item to a bucket that is already full. Chaining works almost like a linked list where it points to the next bucket if the current bucket is full- this is what makes for the linear shaped curves in the plot. Once there is the first collision, it may point to another collision before being able to add an element to the table. In my code, I did not allow for random numbers that were repeats to be added to the table, therefore the load factor is less than 1. If a number is repeated, it is simply ignored, and the function moves onto the next number.

Chaining resolution with key Mod Table Size function



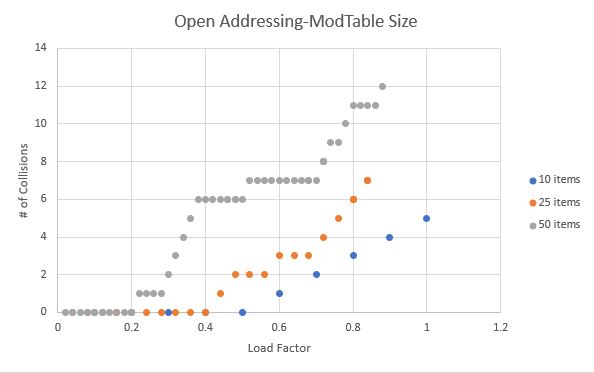
This plot represents a hash table created using chaining with a mod table size key generation method. Rather than the linear trend represented on the plot above, this plot is more scattered. This is because the mod table size function is more efficient than that of mid-square. You can see because there are way less collisions when trying to add values to the table. The fact that the load factor is closer to one here just means that there were less repeat random numbers generated.

Open Addressing resolution with Mid Square function



Open addressing works similarly to adding elements to a list or an array. I created a for loop and parsed through checking each bucket for openings or collisions. The results in this plot are very similar to that of the mid square linear plot above.

Open Addressing resolution with key Mod Table Size function



This mod table size function plot looks very similar to the one above as well. The effect open addressing vs chaining is very small compared to the effect of the hash functions used when in context of reducing collisions.