Nicholas Ionata

COP3530

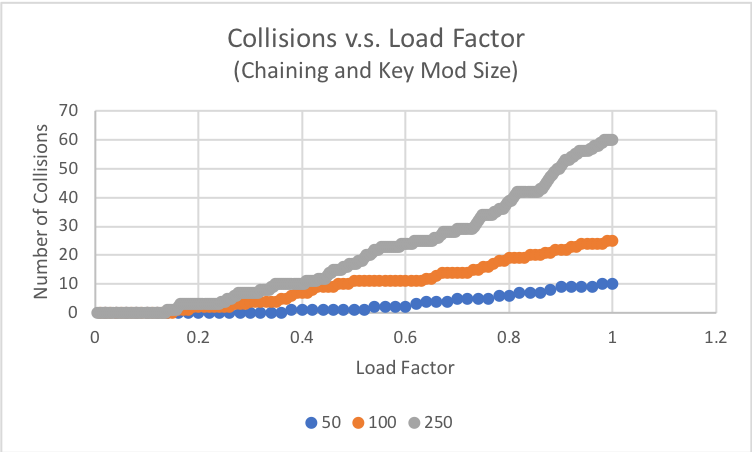
Professor Resch

10/28/18

Hashing Analysis Report

Experiment 1

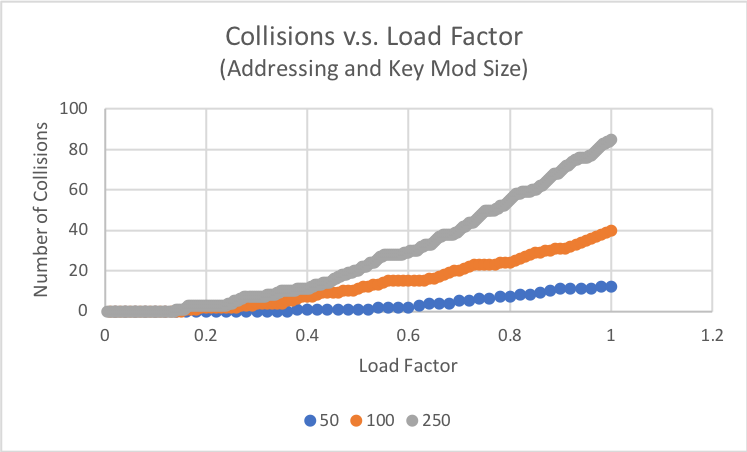
* Hashing Function: Key Mod Size
* Collision Resolution Scheme: Separate Chaining



* From this plot, we can see that the number of collisions increased linearly as the load factor increased. When it comes to the table size, again the trend line scaled linearly as the table size increased by a factor of two and five from the original size. These trends show that this combination of hash function and collision resolution scheme is a strong choice for a hash table as all parts of it scale linearly: increasing the table size and the load factor.

Experiment 2

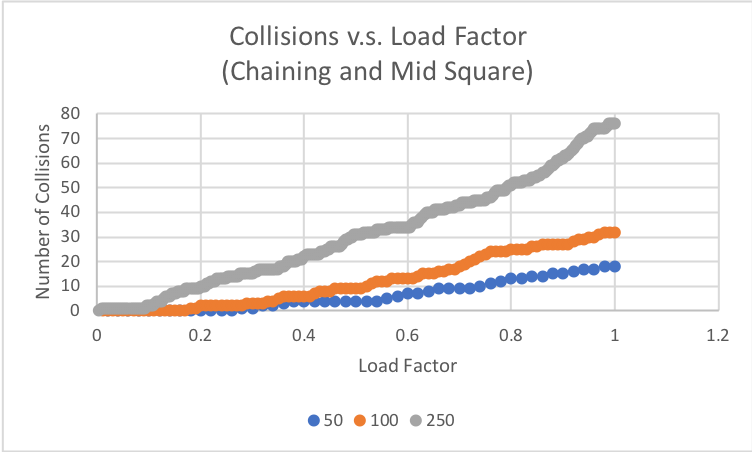
* Hashing Function: Key Mod Size
* Collision Resolution Scheme: Open Addressing



* From this plot, we can see that the number of collisions increased exponentially as the load factor increased. When it comes to the table size, the trend line scaled exponentially as the table size increased by a factor of 2 and 5 from the original size. These trends show that this combination of hash function and collision resolution scheme might not be the best choice because as you increase the table size or the load factor, the number of collisions increase exponentially.

Experiment 3

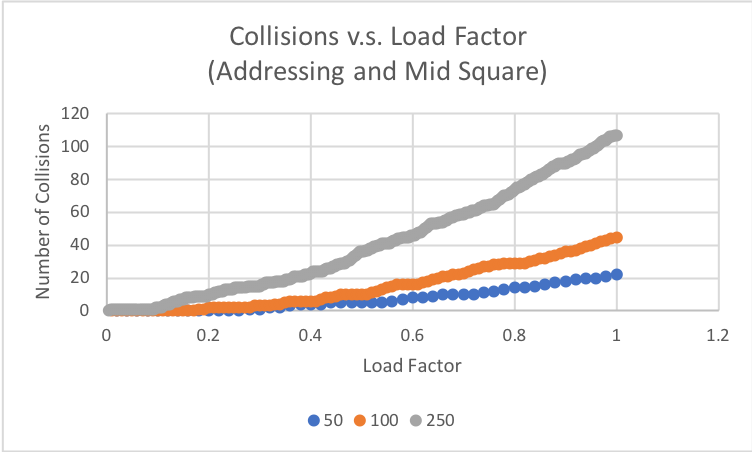
* Hashing Function: Mid Square
* Collision Resolution Scheme: Separate Chaining



* From this plot, we can see that the number of collisions increased linearly as the load factor increased. When it comes to the table size, again the trend line scaled linearly as the table size increased by a factor of two and five from the original size. These trends are very consistent with Experiment 1, however, there are a greater number of collisions. That being said, this combination of hash function and collision resolution scheme is a decent choice as all parts of it scale linearly: increasing the table size and increasing the load factor. The only thing to consider is more collisions occurred in this experiment using chaining than the other experiment using chaining.

Experiment 4

* Hashing Function: Mid Square
* Collision Resolution Scheme: Open Addressing



* From this plot, we can see that the number of collisions increased exponentially as the load factor increased. When it came to the table size, the trend line scaled linearly as the table size increased by a factor of 2 and 5 from the original size. These trends are consistent with Experiment 2, however, the table size scaled linearly and there are a greater number of collisions. That being said, this combination of hash function and collision resolution scheme is not be a good choice because as you increase the load factor, the number of collisions increase exponentially. Additionally, more collisions occurred in this experiment using open addressing than the other open addressing experiment.

Collision resolution schemes on collisions versus load factor

* To compare the collision resolution schemes and how they affect collisions, we can look at Experiments 1 and 2 or Experiments 3 and 4 as the hash functions are held constant. In both of these experiments, it is pretty clear that as the load factor increased, the number of collisions increased linearly for chaining and exponentially for open addressing. Based on this trend, chaining is the better collision resolution scheme. With smaller table sizes and less filled tables, open addressing might beat chaining in collisions, however, as the load factor increases (the table fills up) and the table size increases, separate chaining will result in a lower number of collisions. Taking this even further, when comparing the experiment pairs, chaining performed better with the key mod size hash function than mid square. The only thing that must be considered is the additional memory that chaining takes up.

Hash functions on collisions versus load factor

* To compare the hash functions and how they affect collisions, we can look at Experiments 1 and 3 or Experiments 2 and 4 as the collision resolution schemes are held constant. Both experiments in the pairs had the same rate of change, linear and exponential respectively. The comparison that can be noted is the overall number of collisions. Mid square hashing sees 10 to 20 more collisions than key mod size. Based on this trend, it is fair to say that the better hashing function would be key mod size as it produced less collisions overall, and as the load factor increased. Taking this even further, when comparing the experiment pairs, key mod size performed better with chaining than addressing.