Nicholas Miller

Hash Analysis

**Plot Discussion**

The open addressing resolution schema results in an exponential growth of collisions vs load factor. This is inpart due to the fact that as the load factor increases the avalablity and chance of a certain position being open decreases significantly. This results in many collisions when attempting to find an open position in the table. The gorwth rate of open addressing is poor in collision resolution. Additionaly, as the size of the table increases so does the number of collisions so there is a proportionality between the size of the table and the number of collisions.

**Plot Discussion**

The chaining resolution schema results in a linear growth of collisions vs. load factor. This is inpart due to the fact that keys can be placed in the same position in the table because at each poisiton of the table lies a linked list allowing for an unlimited amount of inserts. This gives great results when inserting keys into a hash map. Additionaly, as the size of the table increase the slope of the collision vs load factor increases at a slow rate. In all, Chaining has great results and is a good method for a collision resolution schema.

**Collision Resolution Schema**

After comparing the two resolution schema’s, it is clear that chaining is a better schema for the reduction of collisions. Without regards to hash function, chainging clearly out shines open addressing for one major reason. That is for open addressing, as the load factor increases the number of open positions in the hash table decreases. Meaning that as the load factor increases the chance of a positions being open is greatly reduced. So each insertion would likely result in more collisions when looking for an open position in the table. On the other hand when chaining, positions in the table are garenteed to have a open postion because at each positon in a chained hash map lies a linked list. This design allows you to keep inserting elements into the load factor grows past one because each element will always have a position in a list inside one of the positions. In all, chaining produces a significantly smaller number of collisions.

**Hash Function**

After comparing the results from both hash functions, I noticed that there wasn’t a large difference in the abilty to distibute the keys equal across the table. I believe these results came to be because the keys were randomly produced resulting in a equal distribution of key values. In these cases the hash function doesn’t play a large role in the determinination of position meaning both functions have similar results. If the keys werent randomly distributed a mid square hash would have much better results, but since this experiment had a random distribution of keys the results were the same. In all, this experiement produced similar results in collision vs load factor because values were randosmly distributed.