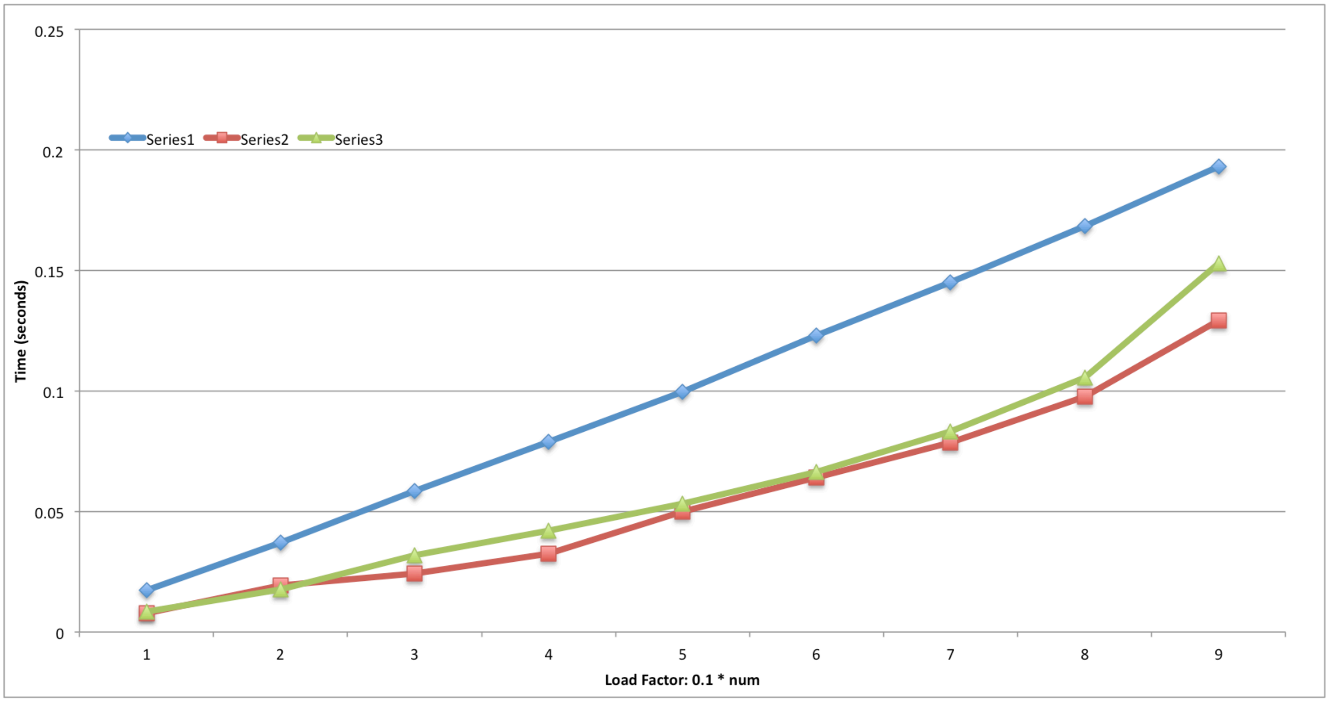
Tim Elvart
2760606
2/19/17

EECS 560: Lab 4
Comparing Hashing Methods

For this lab, I decided to make a class named “Test” that would be handling all of the timing for the different load factors and hashing methods. I started with a function that will time how long it takes to insert (loadFactorBeingTested) * (tableSizeDesired) keys into a hash table using each of the different hashing methods. It was done that way so all three tables could use the same seed for the random number generator. A function was then made that would average the results from each of those tests over 5 trials, for a given load factor. This ensures that each trial had a different seed for the random generator. The data was obtained by gathering the results of the average-r function, for the load factors 0.1,0.2,…,0.9, with a table size of 600011. The keys are in a range of 1 to 2,147,483,647. The results of this test are shown in the graph below.



In this graph, “Series1” represents a table using an open hashing scheme, “Series2” represents a table using the quadratic probing scheme, and “Series3” represents a table using the double hashing method. As seen from the graph, the two tables using a closed hashing scheme significantly outperformed the table using open hashing. The time it took for the open hash table followed a more linear scheme as the load factor increased, while as the two closed hash tables followed a more exponential curve. The machine my test program executed on is a 2-core MacBook that was released in early 2008. If the test program were to be run on a more powerful machine, I would expect faster times in general for all of the data, however I would still expect a tables’ performance to follow the same curve.