// Dion Niazi dn3gy 16 03 2017 postlab6.pdf

The big theta running time for my application is n^3. The reason this is so is because the rows and columns is not a constant so the big theta run time is n^2 and the worst case scenario for my hash table is n so combined together I get big theta n^3.

Timing for my lab was really good from the beginning when I did the pre-lab. Timing was all done on my MacBook and not on the lab computers so I have no idea whether my code is faster or slower from the lab computers. In the pre-lab my code when complied with the –O2 flag, my program ran in under 3 seconds (for the 300x300.grid.txt and the other ones). For the 250x250.grid.txt though, it wasn't much different just 4 seconds running time. Because my pre-lab code was so fast, there wasn't much to optimize has I have already met the basic requirements for the lab. I did though fixed my code where I changed my hash function slightly and buffered my output. I stored the output in another array and printed after the timer stops and my 250x250.grid.txt time was somewhere between 2-3 seconds. Looking at the 250x250.grid.txt file my overall speedup optimization is about 1.33 (the optimized time being 4 seconds and the optimized being 3 seconds, then 4/3 = 1.33).

The new function I used was index = the first character of string % table size. This was worse because more collisions occurred as many strings may have the same first character. Time was 4494 milliseconds, which was surprisingly fast to me compared to what I thought. The table size to make performance worse that I chose was 2 and the time I got was 49451 milliseconds, which is horrific. It was worse because everything hashed to only 2 buckets so it made 2 lists with n time. Both were tested with the 300x300 grid file.

No problems were really encountered in the post lab other than not being able to optimize so much because of already optimization. I didn't try linear probing because I know for a fact that it would be slow so I just stood with separate chaining.