**Coding Assignment 6 Results**

CSE 3318

**Test 1**

**A.** How many rows are in your file/how many cells are in your hash table array?

21

**B.** How many of those cells contained the head of a linked list?

30

**C.** What percentage of the array is being used?

*70%*

**D.** What is the length of the longest linked list?

*4*

**Test 2**

**A.** Did increasing the size of the hash table array give you different results than Test Question 1?

*Slightly, as the amount of lists increased from 21 to 25 and with the longest list containing 3 entries instead of 4.*

**B.** Explain why or why not.

*By changing the HASHTABLESIZE, you also change the result of the remainder or modulus of the Hash and the more amounts of differing results.*

**Test 3**

**A.** How many rows are in your file/how many cells are in your hash table array?

*14*

**B.** How many of those cells contained the head of a linked list?

*15*

**C.** What percentage of the array is being used?

*93%*

**D.** How did decreasing the size of the hash table array affect the percentage of the array that filled?

*By reducing the size of the hash table less than of the Input file size, the chances of the array filling up are higher as there are more files than the spaces available itself.*

**E.** Did your hash table get any linked lists that were longer than in Test 1? Why or why not?

*Yes as the longest list now contains 5 entries. This is due to the lower amount of space and with the Table size being less than the file size, it increases the chances of an entry being placed in the same array.*

**Test 4**

**A.** What was your average search time when your HASHTABLESIZE matched the number of records in the file?

*2.1 tics*

**Test 5**

**A.** What was your average search time when your HASHTABLESIZE was set to 1?

*1.1 tics*

**Test 6**

**A.** What was your average search time when your HASHTABLESIZE was set to 1 and you only searched for the last record of your input file?

*1.4 tics*

**B.** Was this average different from your answer to Test 5. If yes, why?

*Slightly yes, as the average went up by 0.3 tics and it is since it had to search through the entire list to find the last entry.*

**Bonus Question**

If your program was using Open Addressing rather than Separate Chaining, then how many cells of the hash table array would be used when HASHTABLESIZE is set to the number of lines in the file? How did you calculate this number? Show/explain your answer.

*With using Open Addressing and the HASHTABLESIZE set to the number of lines in the file, it would be a total of 30 cells being used, with the entirety of the hash table being used. This is because with open addressing, you cannot occupy a cell that contains an entry and if an entry tries to do so, it will move on to the next available cell. And with the HAHTABLESIZE set to the amount of lines in the file, its guaranteed that no entry will collide with another.*