**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?

Ans:30

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

Ans:19

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

Ans: 63.33%

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

Ans: 4

**Test 2**

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

Ans: Yes

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

Ans: Increasing the size of the hash table while keeping the number of entries in the input file same decreased the density of the hash table as now there were 23 filled rows out of 60 which has a percentage occupancy if under 50%. Also, the length of the longest linked list decreased from 4 to 3.

**Test 3**

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

Ans: 15

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

Ans: 12

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

Ans: 80%

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

Ans: As the number of cells decreased, the density of the table increased so more conflicts arose, so the percentage of the array filled increased.

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

Ans: Yes, we got linked lists that were longer than in Test 1 because more conflicts arose, so the contents of the file with same hash index had to go to next linked list.

**Test 4**

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

Ans: 2+1+0+2+1+1+2+1+1+2 = 13

13/10 = 1.3 average

**Test 5**

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

Ans:1+1+1+1+1+1+1+1+1+1 = 10

10/10 = 1.0 average

**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?

Ans: 2+1+1+1+1+1+1+1+2+1 = 12

12/10 = 1.2 average

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

Ans: The average was greater than test 5 because when we are searching in a hash table, we are doing a linear search on every index element. Thus, in test 5 when we were searching for random elements, some were found closer whereas the other elements were found far away from the head whereas in test 6, we were searching for the last element repeatedly so were doing a linear search to the very end of the hash table which caused the average time to be higher than that in test 5.

**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.

Ans: Since my hash table contains 30 cells, open addressing uses linear probing, which ensures that every cell is filled.