How did two data structures compare in your tests?

To compare the two data structures, the program was run 10 times for each database (text file) given and once more for a separate database (text file) I created. The database I had created was made to test my program however the resulting test data was interesting when the run time of the two data structures were compared. When testing my program with the database I created only 3 accounts are created prior to "start" and the number of accounts remains between 4-6 after start were each operation supported by the program is used once after "start". The resulting data showed how the sorted array may have a faster run time for smaller database, however this is not the case in the long run.

From the averaged data collected from ten trials per database, when the two structures are compared as the size of the database increase, we can conclude the hash function structure is more efficient in the long run. With a Big O (1) for its search, insert and delete operation, hash tables have a more consistent running time for the given operations. Although it may not be the fastest running time with smaller database, its constant running time makes it the faster running time in the long run.

To create a better visual representation of the data collected to show the change in the running time as the size of database increases, a graph was created of database size is (on the x axis) vs the average running time for the ten trials per database (in nanoseconds) (on y axis) . Looking at the running time for a small database, the sorted array has smaller running time. However, as the size of the database files increase, it can be concluded that as the size increase so does the running time of the sorted array while the hash table maintains a rather more consistent running time. More generally, in conclusion the running time of the hash structure is faster than the sorted array structure when performing 100 000 operations for each database as the asymptotic category of hash tables falls under O (1), which is a constant unlike the sorted array.

Is one significantly faster than the other?

Yes, as the database size increases so does the running time for the sorted array while the running time for the hash table remains at a rather consistent value. As the size increases so does the average time difference between the two data structures due significant increase in the running time for the sorted array while remaining constant for the hash function. This shows us how the hash function is significantly faster because it maintains a consistent running time despite the data size.

Did you expect the results that occurred? Why?

Yes, the theories discussed in our lectures and the charts we have analyzed for the big O of both data structures provided a general knowledge and understand of the result we are expected to acquire.

What do you predict the results would be if you increased the data base size to two million?

The result for the Time to compute 100 000 operations regardless of the data base size would remain fairly consistent for the hash function while it would continue to increase for the

sorted array. Regardless of the size of database the theory remains intact, meaning the hash table function would continue to have big O (1) for all its computations while the sorted array would have O(n) for its insert and delete function and O(lgn) for its search.

}