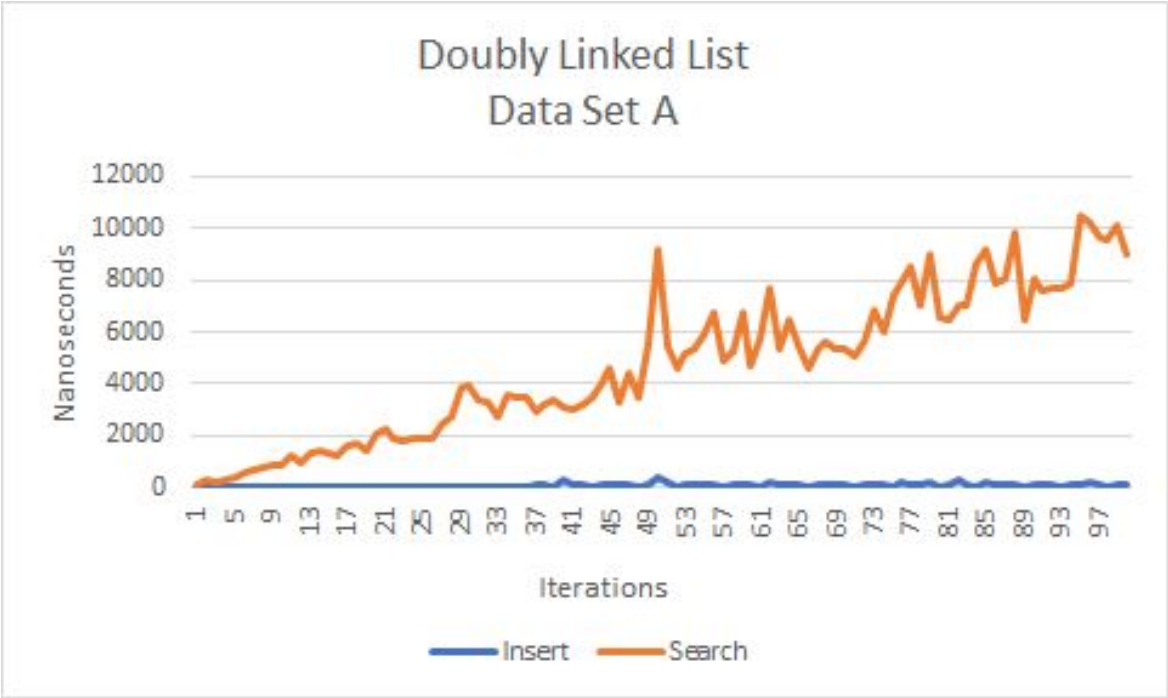
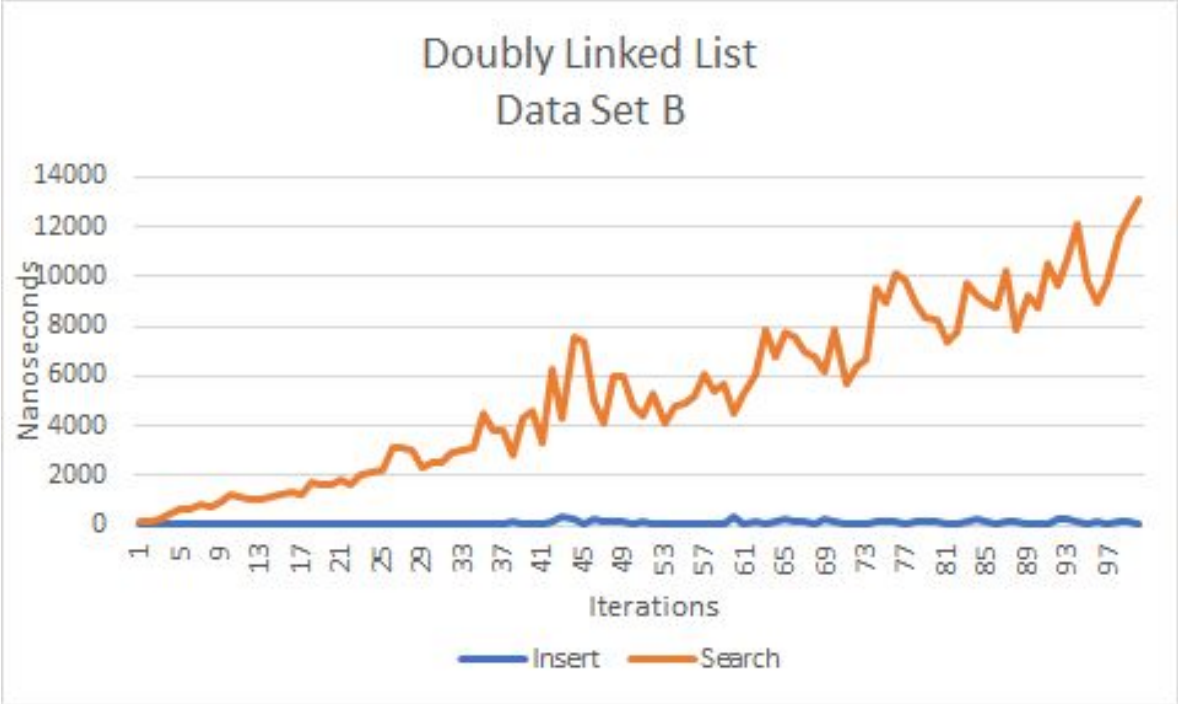
CSCI 2270 - Final Project Report

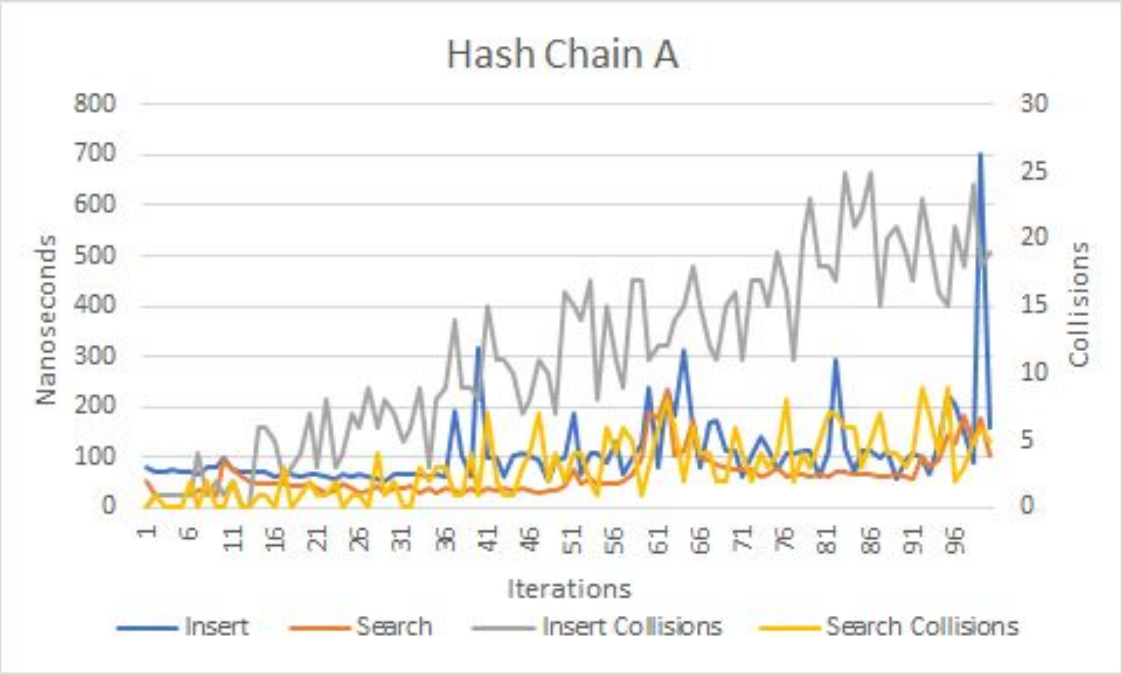
Vera Duong

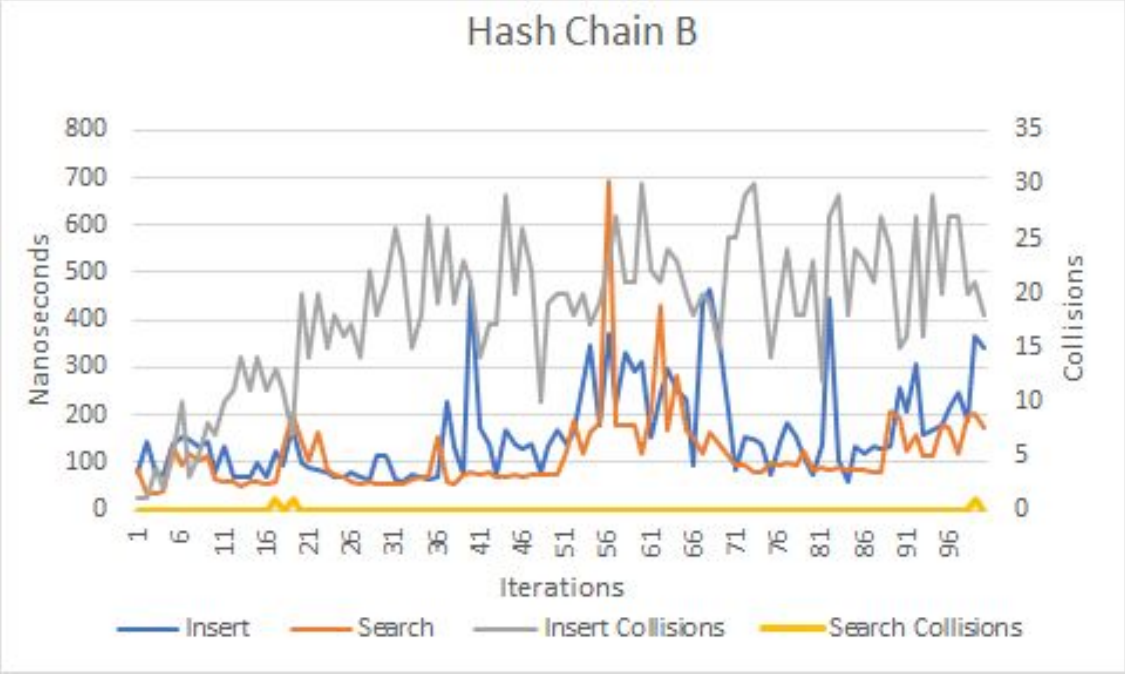
Douby Linked List Figures:



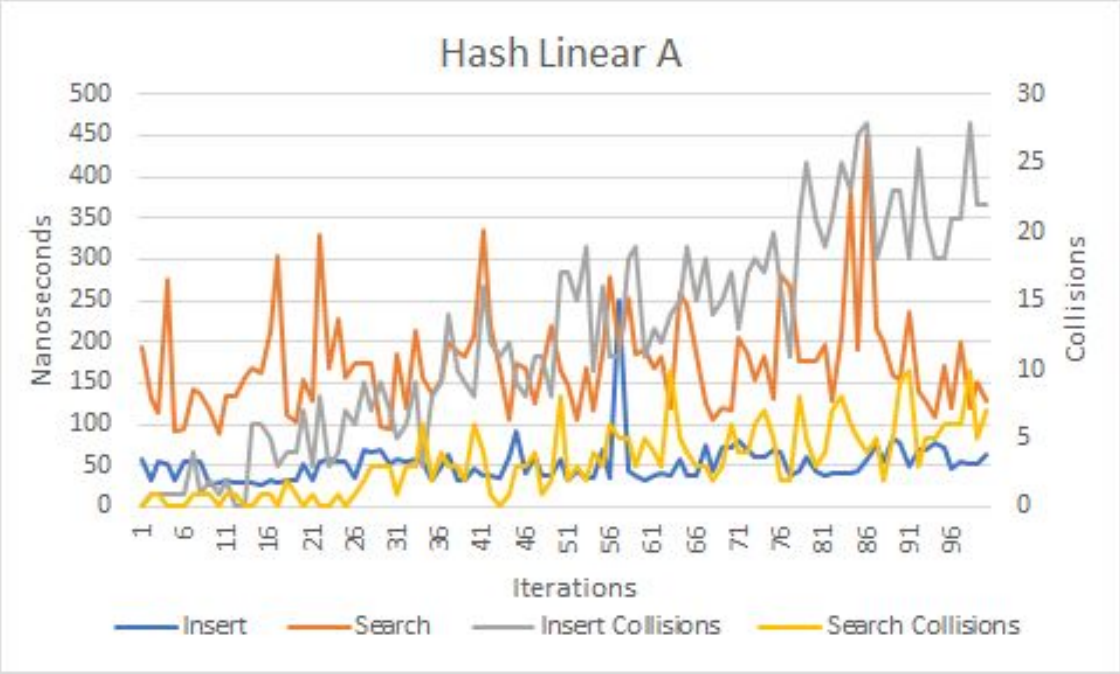


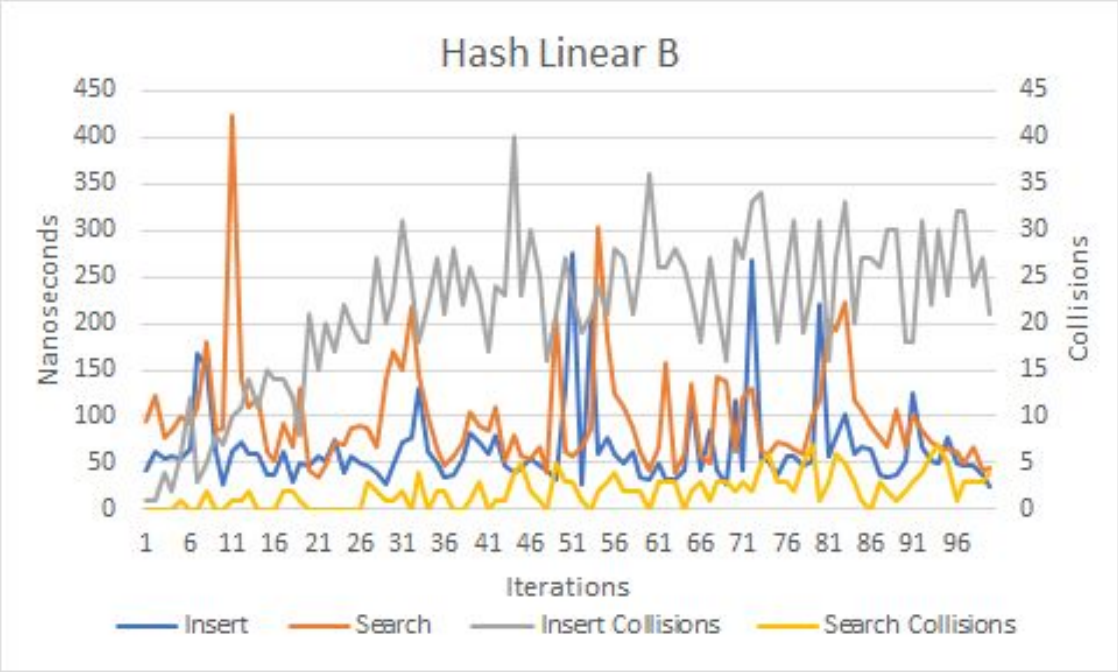
Hash Chaining Figures:



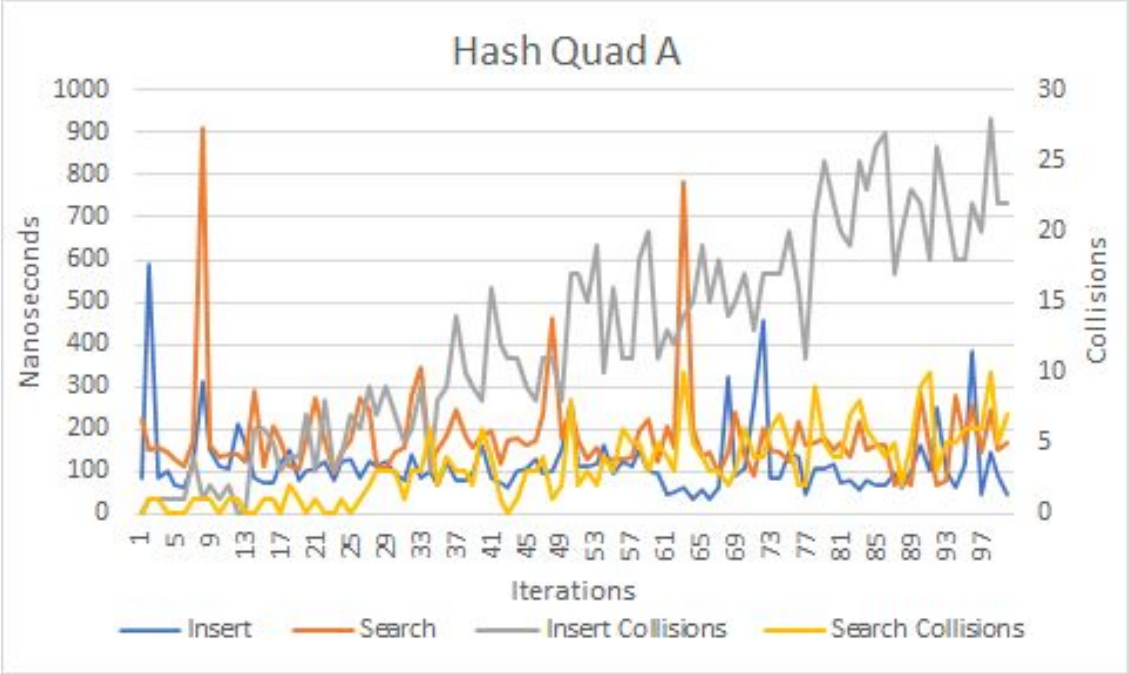


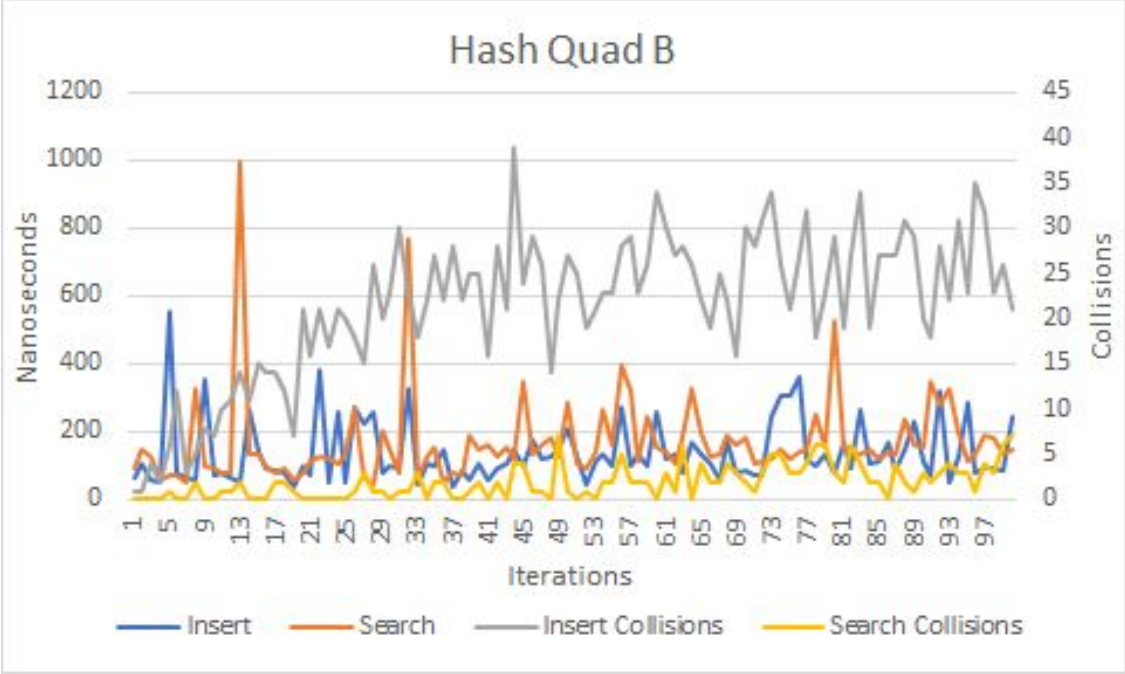
Hash Linear Probing Figures:



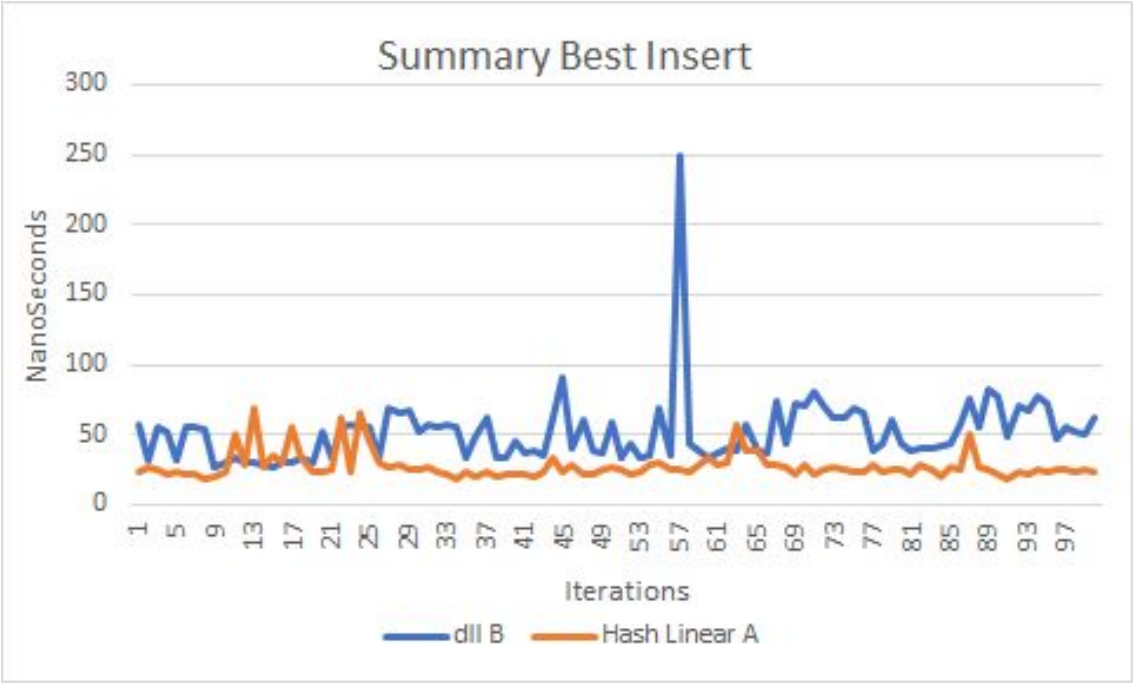


Hash Quadratic Probing Figures:





Summary Figures:



Summary:

In this project we needed to store patient IDs in a data structure using hash tables. We needed to implement and test different methods of hashing, which were doubly linked lists, hash chaining, hash linear probing, and hash quadratic probing. For the doubly linked list, we needed to test the time complexity for inserting and searching. For the other three, we needed to test insert, search, insert collisions, and search collisions.

If we look at the doubly linked list figures, we can see that the search increases (which matches the example doubly linked list figure in the write up) but the insert doesn't and looks linear. It may be because for the insert, we are simply adding the key to the tail and there isn't a traverse which would make the graph increase. Furthermore, seen in the graphs, linear probing was best for inserting and quadratic probing is second best. I expected the graph for linear probing to have more spikes because it's more prone to clustering, but it all depends on the data, therefore quadratic probing had more spikes that may have been caused by secondary clustering. This is because elements that hash to the same hash key will usually probe the same alternative buckets. The best method for search was hash chaining, and the second best is linear probing. It's shown that searching for chaining was faster than the other data structures, which I didn't expect since linear probing has the best cache performance out of our two open addressing methods. But, chaining insert is slower than the other open addressing methods, this is because its cache performance is poor, and it shouldn't be used to store patient IDs because some buckets of the hash table are never used which leads to wastage of space.