Report on final project

In this final project, we are asked to write a code using different data structures to insert or search. My hypothesis is that Hash table should be the fastest data structure to dealing with those data. Linked list’s time is depends on the length. Once knowing that the data will be huge it will be low efficient to using the Linked List. As for binary search tree, it depends on the data that you trying to plug. Once the data was very small, it only takes a very short amount of time. But if the data key was a super huge, it needs to be reached to the end of the tree. And I think it would take more time than Hash table. However, in the experiment that we find out that the binary search tree adds faster insert time than any other data structure, that we have seen in our experiment and the insert time was not linear it would not grow as the data gas more. But it depends on how the key data are. Overall, the performance on hash table search and binary search tree search is faster than Linked List. Quadratic probing is extremely slow specially on a large data set. It can go on for minutes if the collisions are not resolved. In this case where the data set is so huge, it will keep on checking for a free slot and most of the times it will not be empty. The traversing will continue until a free slot is found. This is the con of quadratic probing. It is quite slow. And because application needs, USPS is required lots of searching instead of plugging I would recommend them to use binary search tree or hash table due to the fact that they are doing the fastest searched among other data structures that we have used in this experiment.















