University of Colorado - Boulder

CSCI 2270 - Data Structures

Final

Final Executive Summary

Authors:
Cole MacPherson[a]
Scott Mansfield[b]

[a]SID: 108521329
[b]SID: 109029607

Instructors:
Trivedi

April 29 2020

I. Findings

To help out the USPS with the bottle-necking of their mail tracking software, multiple data structures were analyzed for their insertion and search times. The first step in this process was to view how the tracking IDs were sorted between the two sets that were provided. By doing this, it was found that the first data set had random tracking IDs throughout the file and the second data set had the tracking IDs increase as the file index increased. This suggests that there is some sort of order in the way the tracking IDs are generated. This trend can viewed in figure 2 and the random generation of tracking IDs throughout the first data set can be visualized in figure 1. Through testing, it was found that a linked list was the worst-performing data structure in both the search and insert tests. Figures 3 and 4 show the large time values for insertion and search of both data sets for the linked lists. These large times can most likely be attributed how a linked list is structured, as it must traverse the whole linked list to insert and search data at the end. The testing further revealed that the chained hash table was the best performing hash table as it provided the lowest insert and search times. The insertion and search collisions were also considerably lower than the other hash tables. When viewing figure 7 and 8, it can be seen that the chained hash table benefits from the second data set as the time for insertion and search is more consistent throughout the test. In data set A, the random tracking IDs create search collisions for the chained hash table suggesting that chained hash tables benefit from ordered data. The other hash tables suffer from higher insert and search times as the last pieces of data are reached. These hash tables search and insertion times along with their collisions can be viewed in figures 7 through 12. With this information, the conclusion was made that the chained hash table was the best implementation of all the hash tables and it will be used moving forward. The binary search tree was found to be a middle ground between the chained hash table and the linked list. It's insertion and search times were notably better when the first data set was used suggesting that the binary search tree benefits from random values rather than sorted values. The binary search tree data can be viewed in figures 5 and 6. When comparing the insertion and search times for each data structure, it is clear that the chained hash table, once again, is the best performer. The chained hash table has lower insertion and search times than both binary search trees and linked lists. By viewing figures 13 and 14, it can be seen that the linked list skyrockets off of the graph very early and is not close to times of the binary search tree and the chained hash table. From these graphs, it can also be seen that the chained hash table is always below the times of the binary search tree. With the completion of these tests it was found that for the USPS to fix its mail tracking software, a chained hash table should be implemented along with tracking IDs that are ordered similarly to the second data set. This implementation should get the best out of the chained hash table and provide a great improvement from the current linked list implementation.

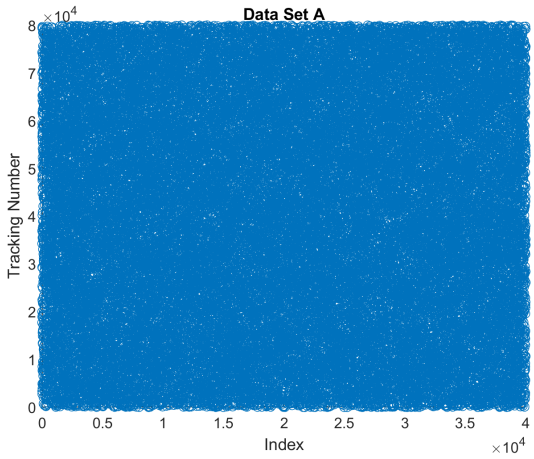

Fig. 1 Data Set A Values

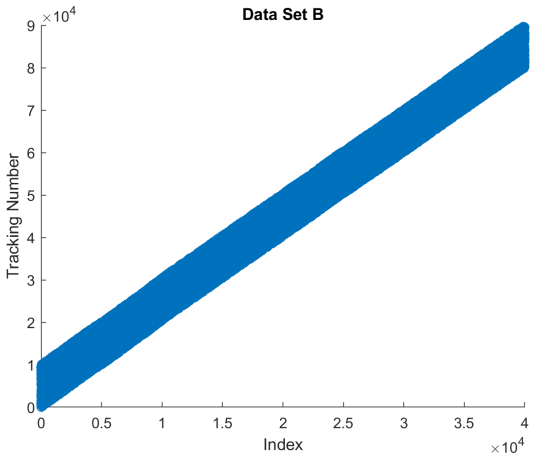

Fig. 2 Data Set B Values

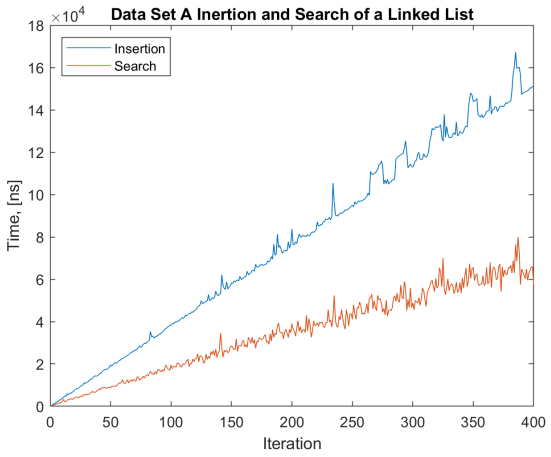

Fig. 3 Linked List Data Set A

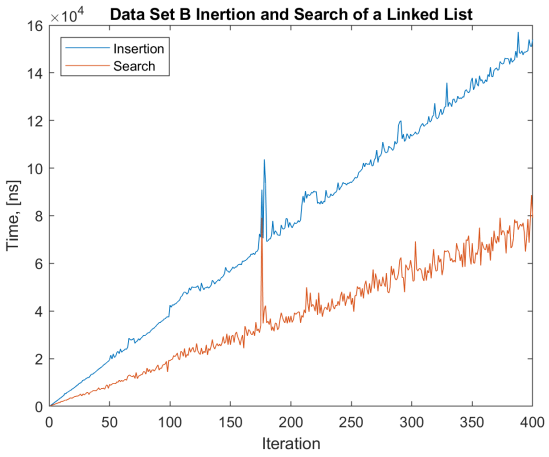

Fig. 4 Linked List Data Set B

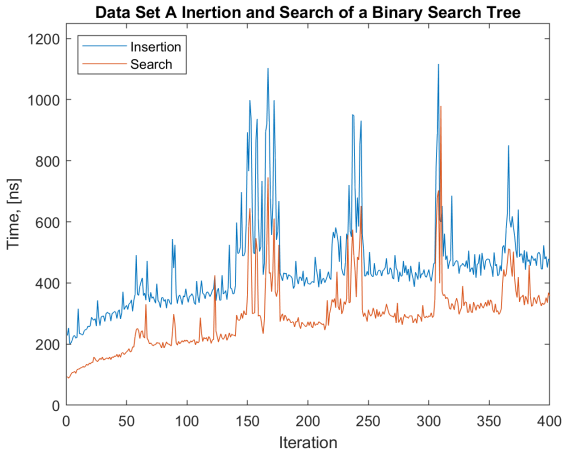

Fig. 5 Binary Search Tree Data Set A

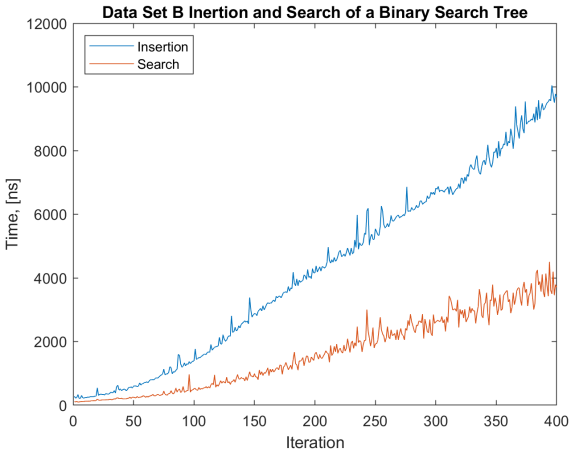

Fig. 6 Binary Search Tree Data Set B

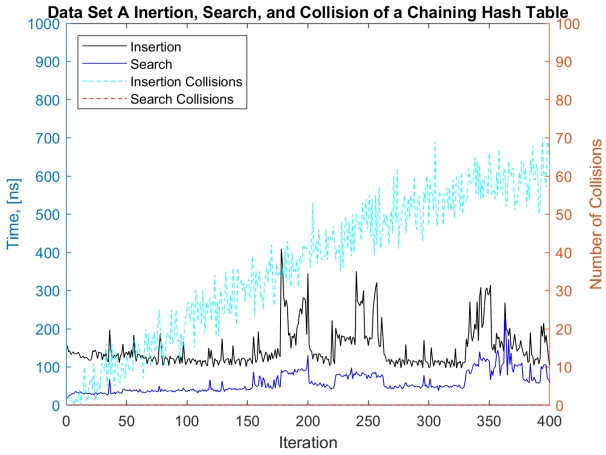

Fig. 7 Chained Hash Table Data Set A

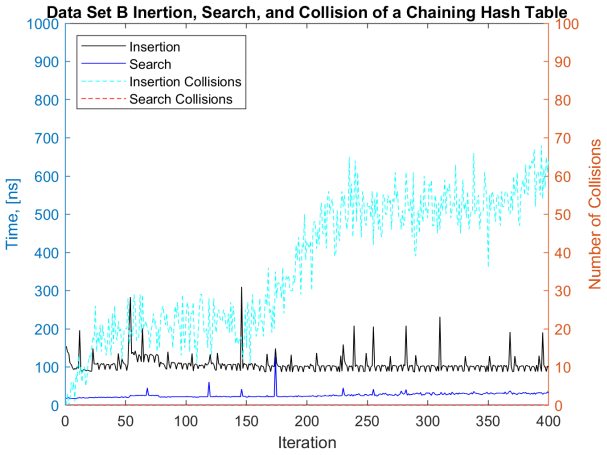

Fig. 8 Chained Hash Table Data Set B

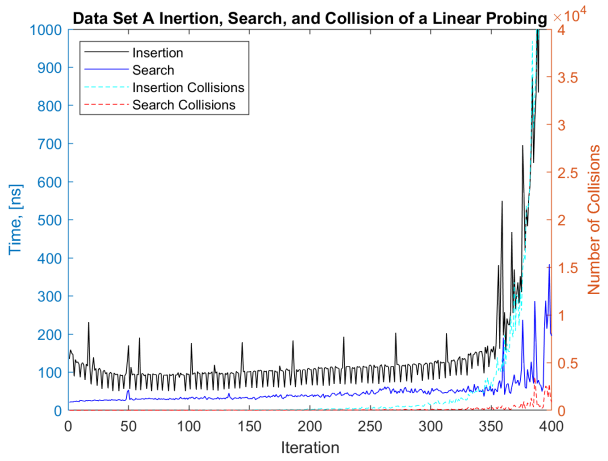

Fig. 9 Linear Probing Hash Table Data Set A

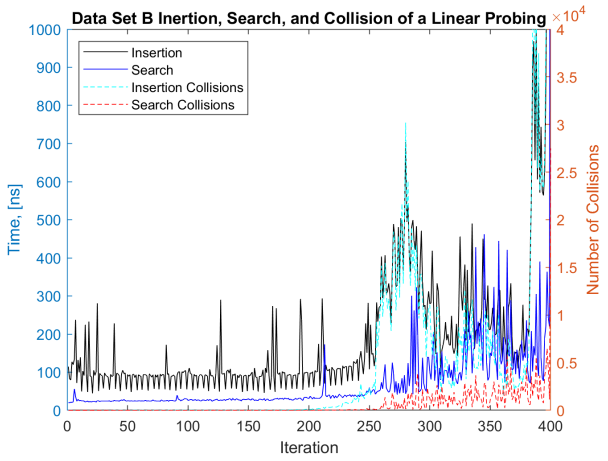

Fig. 10 Linear Probing Hash Table Data Set B

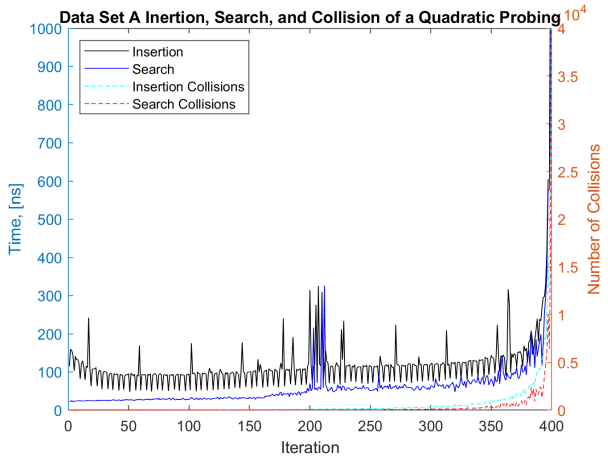

Fig. 11 Quadratic Probing Hash Table Data Set A

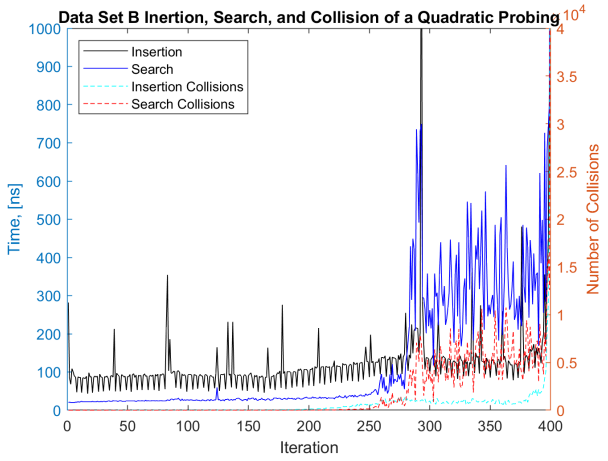

Fig. 12 Quadratic Probing Hash Table Data Set B

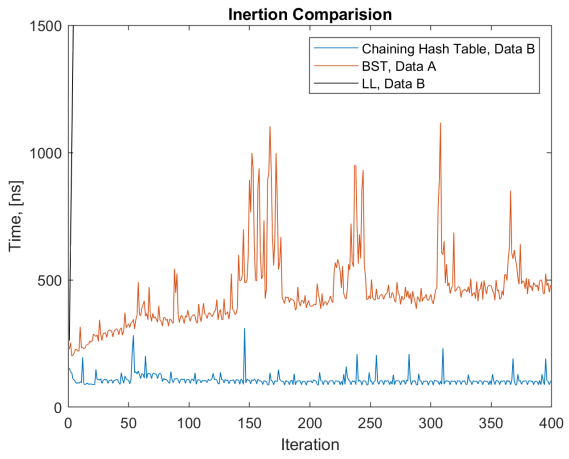

Fig. 13 Insertion Comparison of Data Structures

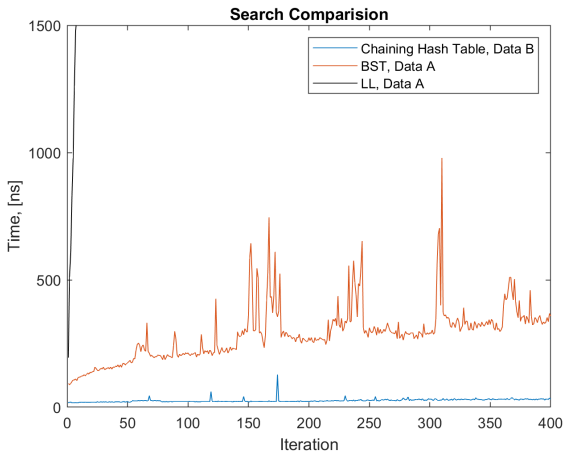

Fig. 14 Search Comparison of Data Structures