Sarah Litz

April 29, 2020

Final Project Report

In executing the different data structures, my findings indicate that the most efficient and reliable data structure for the USPS tracking system is a Hash Table that uses the collision resolution mechanism of quadratic probing. While the binary search tree showed comparable times in its insert execution times, the quadratic probing search time was almost always lower than the tree’s search time, making the hash table with quadratic probing the best option.

A linked list was the obvious worst data structure to use. It showed linear growth (O(n)) for both the insert and search functions, which caused the execution time to become unreasonably large as the data grew bigger.

The binary search tree and hash table with chaining for its collision resolution mechanism had quick execution times for the data sets of 40,000 numbers. However, even within these 40,000 data points, both structures exhibit a slow but steady growth in execution time as the iteration number gets larger. Therefore, since the USPS tracking system works with data sets much larger than 40,000, these two data structures would not be a reliable choice for storing their tracking ID information.

The linear and quadratic probing implementation of a hash table had a similar pattern in which they had very quick execution times, until the end when the array where the tracking numbers were getting inserted was getting closer to being full, and the execution time increased exponentially. However, because this execution time is due to the array filling up, these collision mechanisms are reliable when it comes to big data sets, since they will always have consistently fast execution times until the very end. The quadratic probing implementation manages to beat out the linear collision resolution, because its execution time stays lower for a larger number of iterations.

