Samuel Freed
Asa Ashraf, Sanskar Katiyar
CSCI 2270
29 April 2020

USPS Project Report

Through this project, the big-O insert and search times of the three data types examined were reaffirmed, with the very diverse time values revealing which types are most effective for each set of input data. Firstly, the two open-addressed hash tables are eliminated relative to the chaining table due to longer conflict resolution time and the resulting gradual exponential growth. Then, the comparison of the remaining hash method and the other two types further narrows the options. As shown in figure 12 in the included `reportgen.pdf`, the linked list type produces the best insert times for both data sets, which is expected since insertion for linked lists is O(1); hash tables with chaining follow shortly behind but are slightly longer since they must calculate the hash for each input. However, search times for the linked list make it extremely unfavorable for the post office's use - in figure 13, it increases rapidly, so much so that it is barely visible on the chart. Search times are best for the hash table with chaining, which only requires the calculation of the hash and then index accessing to retrieve a value. As a result, the hash table with chaining would be the most favorable for the USPS's needs. The binary search tree could also be implemented - search and insert are relatively similar in complexity, but are still longer than the hash table with chaining.

One potential issue in this project is the timing of the linked list operations. When compared to the given example, the search times are extremely large and very different from the expected output. I do not think this truly affects the outcome of the experiment, however, since, on average, hash tables have search and insertion times of O(1), whereas the linked list's search time is O(n) on average.