CSCI 2270 Summer 2021 Ashwin Raju
Sec. 301/303

Final Project: Bookstore Delivery Data Management

1. Introduction:

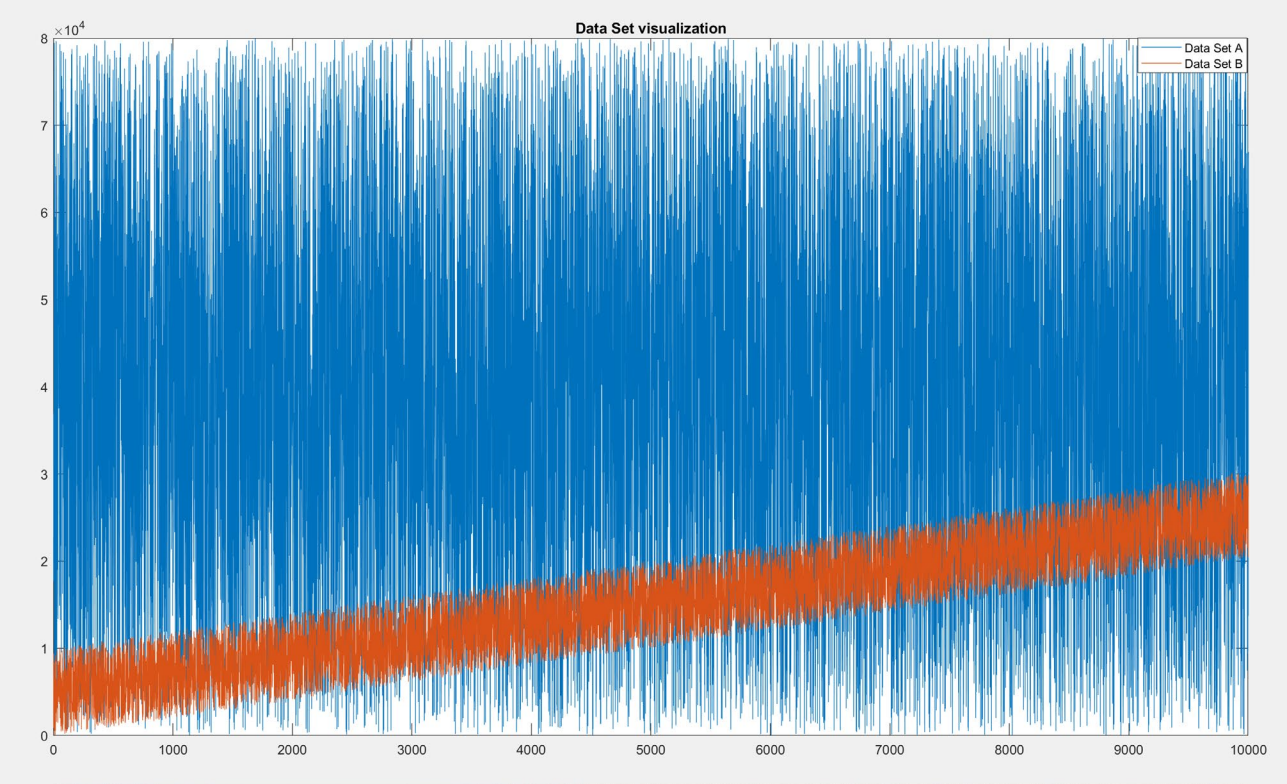
The University of Colorado Bookstore is facing runtime issues in their delivery management system, which currently uses a singly linked list. To resolve this issue, an experiment is conducted to determine the most optimal data structure to manage the delivery data: a singly linked list (SLL), a binary search tree (BST), and a hash table (chaining, linear probing, quadratic probing).

2. Experiment:

The general set up of the experiment is as described in the provided write up, however there are some important points to be aware of. During the insertion of elements into a linked list (singly linked list and hash table chaining), new elements are always added at the head of the list. Also, for the purpose of this experiment, only one collision is counted when a collision occurs, and all subsequent collisions are not counted.

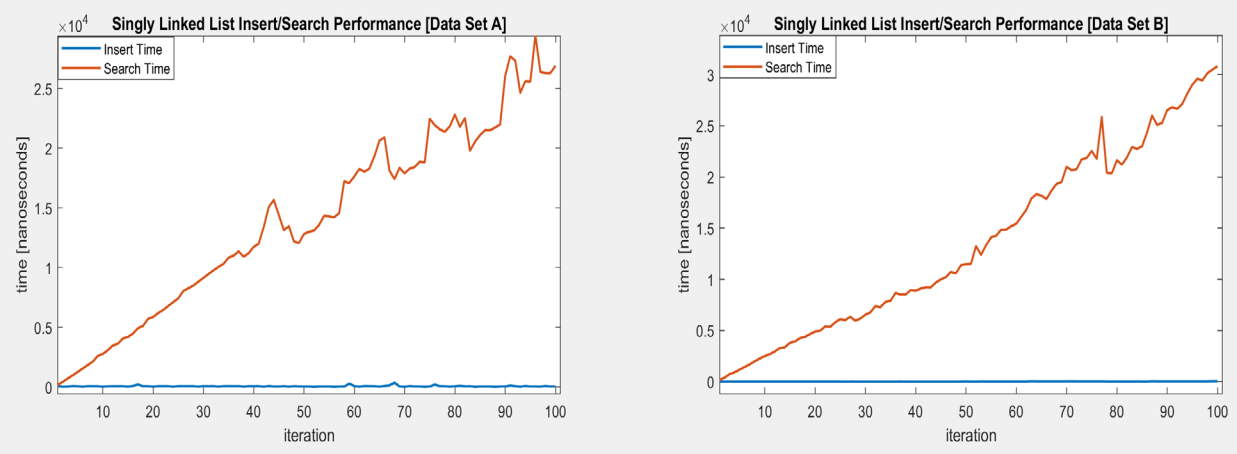
3. Results and Analysis

3.1. Datasets:



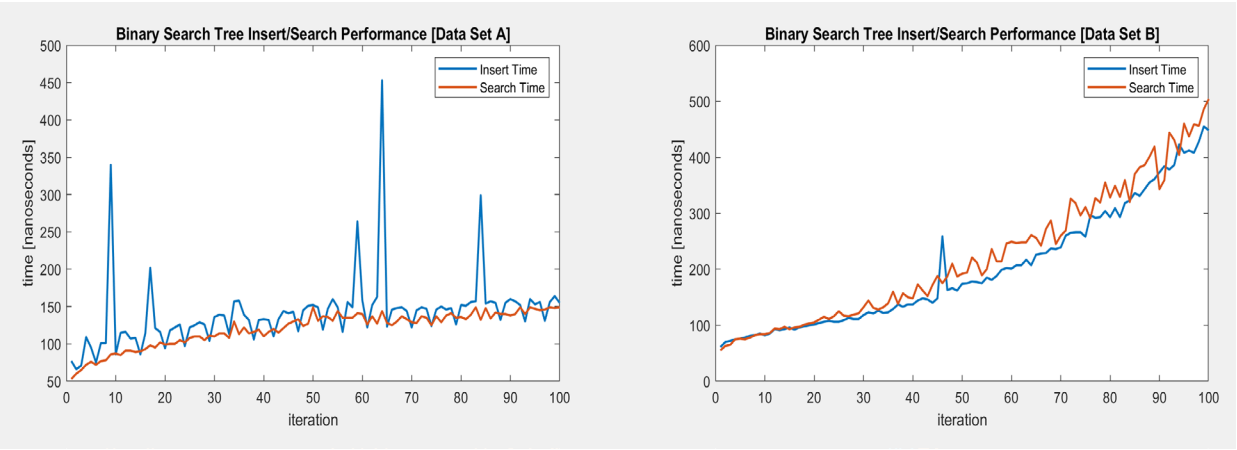
Dataset A is randomized/turbulent and dataset B is linearly ordered.

3.2. Singly Linked List:



Because the insert method adds new elements to the head of the SLL, the insert time is constant with complexity O(1). On the other end, the search time curve is generally linear which suggests O(n) time complexity.

3.3. Binary Search Tree:

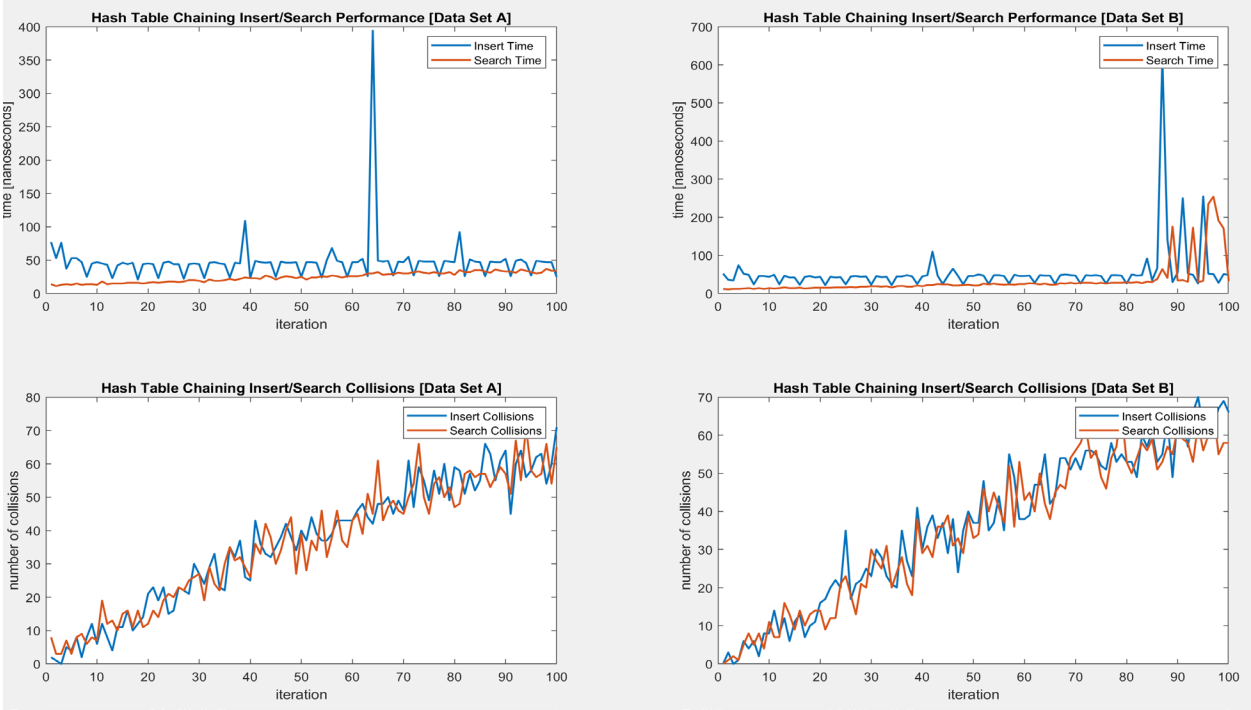


The BST is heavily affected by which dataset it operates on. For dataset A, the BST curve follows the path of a log curve. For dataset B, the curve is relatively linear. This suggests that when using dataset A, the BST functions more optimally and has time complexity of O(log(n)). While for dataset B, the BST becomes a linked list like and has a complexity of O(n).

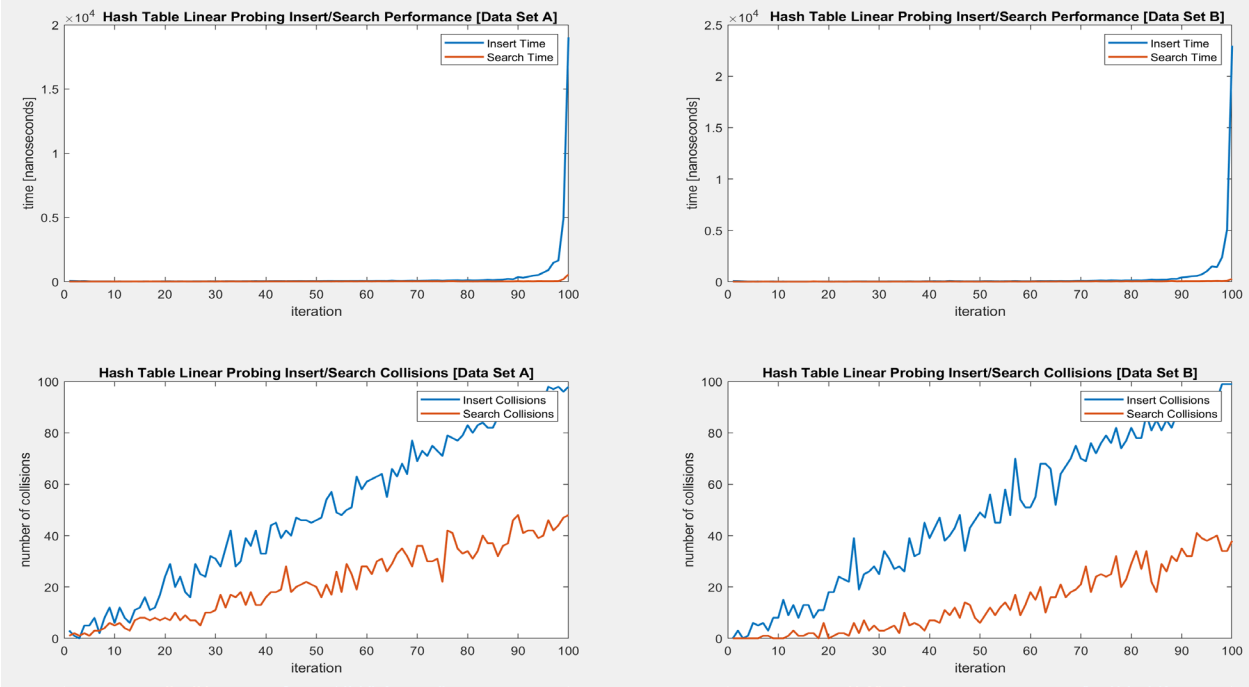
3.4. Hash Table:

3.4.1. Chaining:

The insert times appear relatively constant as new values are always added to the head. The search times also look constant, but upon closer inspection they are linear and have some slope due to the search method traversing a linked list in some instances. The spikes most likely being attributed to collisions for both cases.

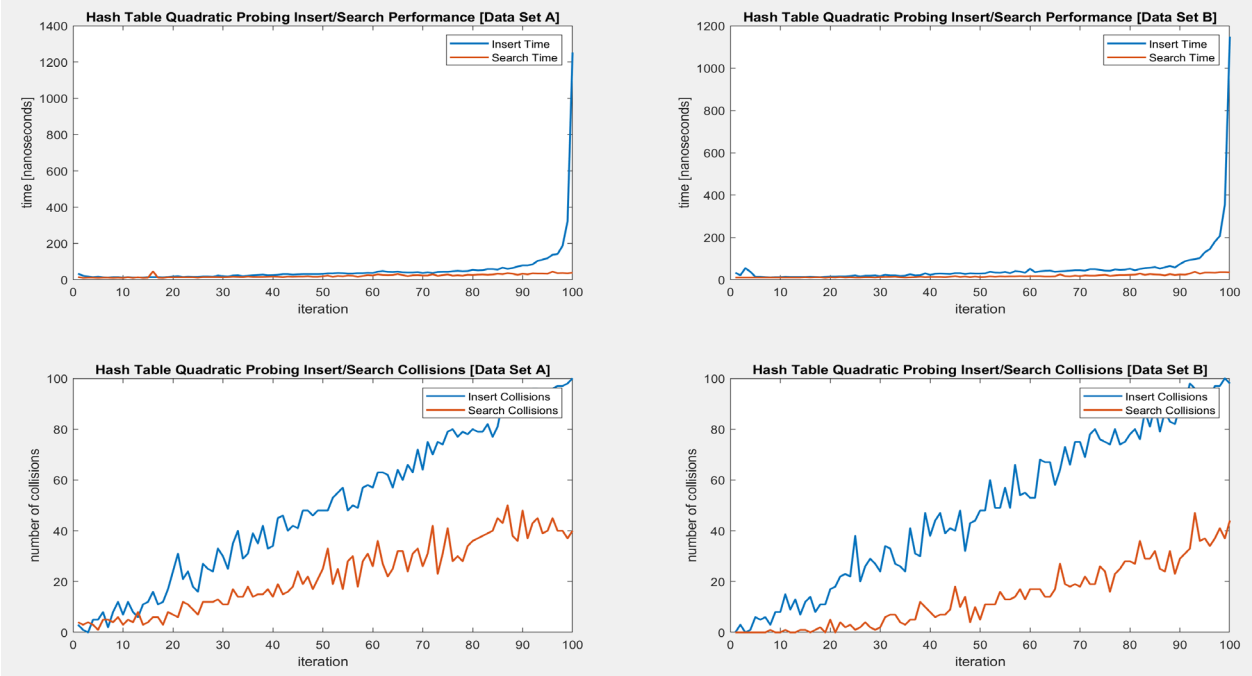


3.4.2. Linear Probing:



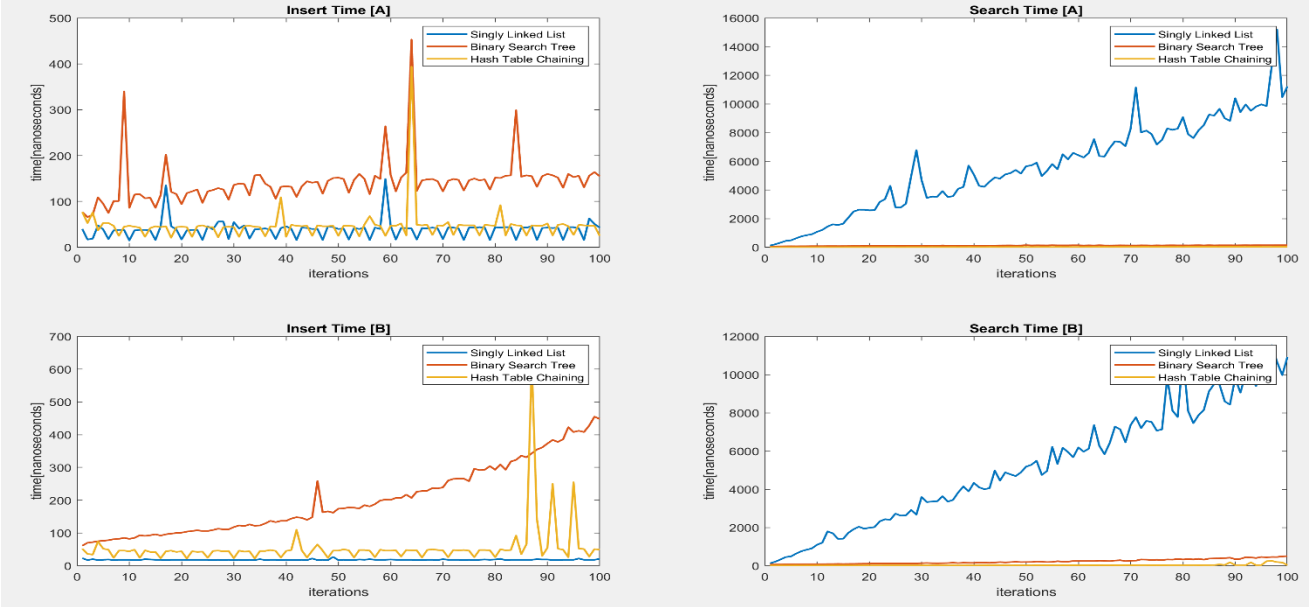
The insert times for both datasets are constant for most of the curve but spike up significantly during the last iterations. This is because as the table is filled, less indices are available for the data resulting in more collisions and a higher runtime.

3.4.3. Quadratic Probing:



The results for a hash table using quadratic probing are similar to linear probing, with the notable difference being the values of the runtime. The quadratic probing results in significantly less than linear probing as there is less clustering.

4. Conclusion:



Reviewing the results, we can eliminate the SLL and BST. This leads to the hash table being the best data structure. In terms of which collision resolution method, although open addressing methods perform well for most of the iterations, their performance deteriorates towards the end of the experiment. This leads to the hash table with chaining being the best data structure. Although it may result in unused space, it overall has the best insert and search times.