CSCI 2270 - Data Structures
CU Boulder
Spring 2020
Final Project - Save The USPS!

Collin Graham
109446646
collin.graham@colorado.edu
TA - Saumya Sinha - 203

Jaryd Meek
109494970
jaryd.meek@colorado.edu
TA - Himanshu Gupta - 101

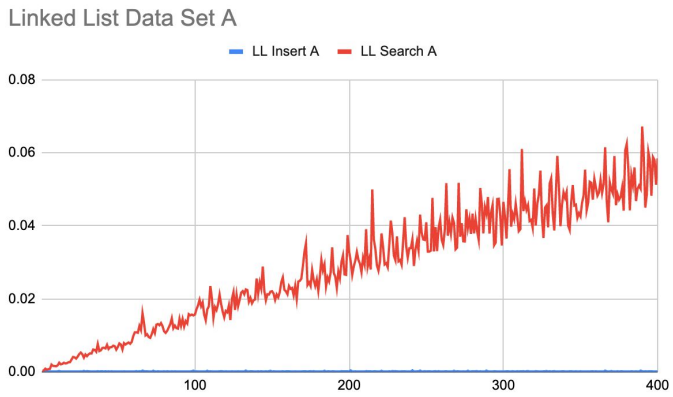
We were tasked with determining the best data structure for use in the USPS Package Tracking Application™. After creating the programs and running all of the tests with the two data sets, we found some interesting results. Firstly, when it came to insertion, the Linked List and Hash Table with Chaining were definitely the fastest. This is because we utilized a tail pointer in the Linked List class, which allowed us to keep track of the end using one pointer and insert the next data point nearly instantly every time (O(1)). This is shown in Figures 1 & 2. The Chaining was likely very fast, as it only had to traverse a very short linked list, but for the hash table with chaining, we appended the new node to the end without a tail pointer.

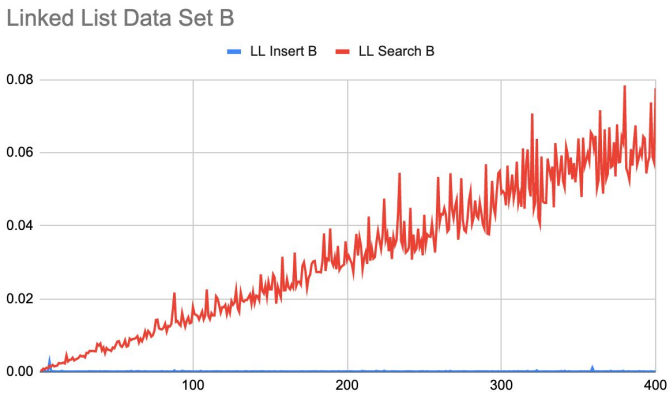
Other notable observations include the fact that both the insertion time and the search time became almost linear for the Binary Search Tree for Data Set B (shown in Figure 4). We believe this to be caused by the fact that the data in Data Set B created a very one sided tree, and in places, likely created a Linked List. When we look at the plot of all the data in Data Set B, this suspicion is confirmed, a one sided tree caused our insert and search times to become linear, instead of logarithmic (what was expected).

For the Hash Tables, we noticed that all three performed better overall when compared to the other data structures, but there were some slight differences between each. The Chaining Conflict Resolution was performing the best overall, then the Quadratic Probing Conflict Resolution performed better. All of the Hash Tables started to increase in time and conflicts, as the Hash Tables began to fill up. For both Linear and Quadratic Probing, the conflicts increased exponentially toward the ends, as they had to find the last few open spots in the storage array (as shown in figures 7-10).

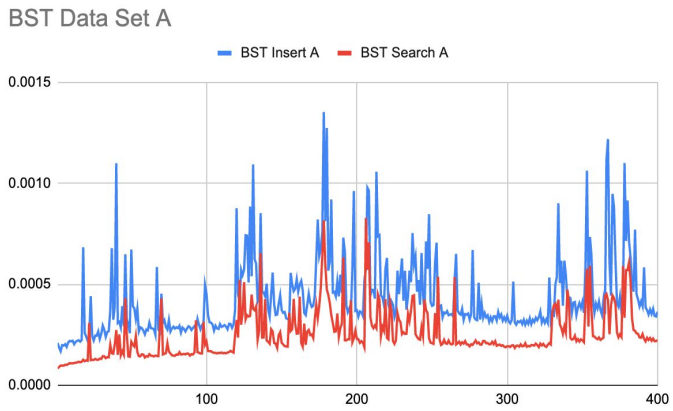
After taking all of this information into consideration, we believe that the best data structure for use in the USPS Package Tracking Application™ is the Chaining Hash Table, because it's performance was ultimately the best throughout most of the tests, meaning it will be the most reliable method.

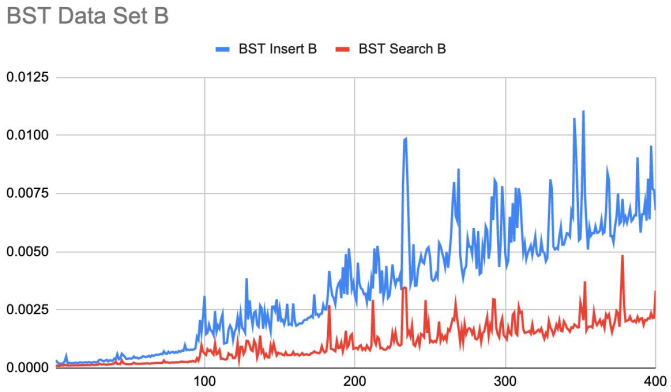
Note - For all graphs below, the Left Y-axis is time in milliseconds and the X-axis is iteration



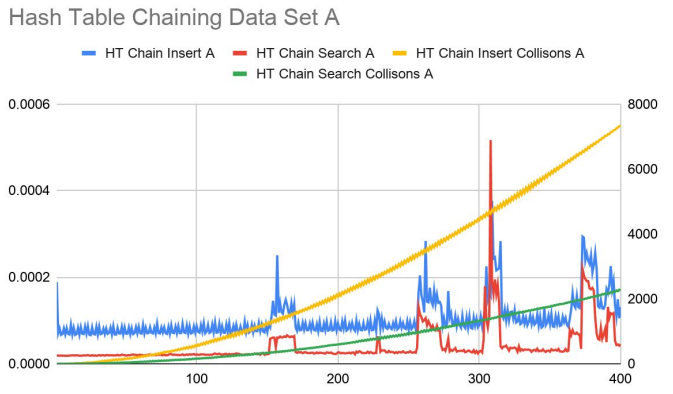


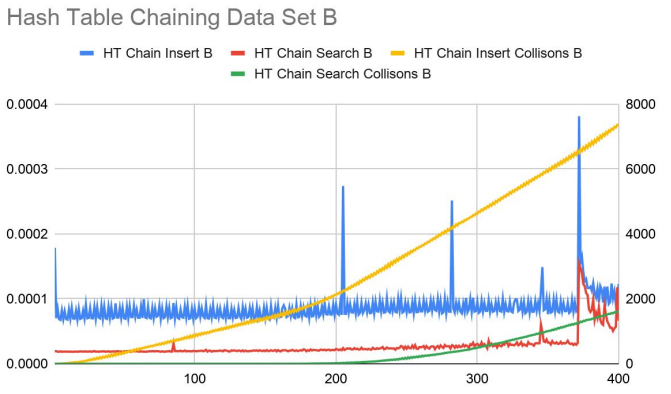
Figures 1 & 2: Average search and insertion times using a Linked List Data Structure for Data Set A and B respectively.



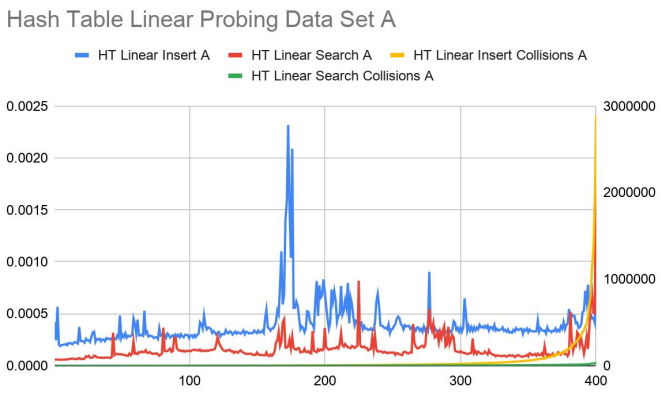


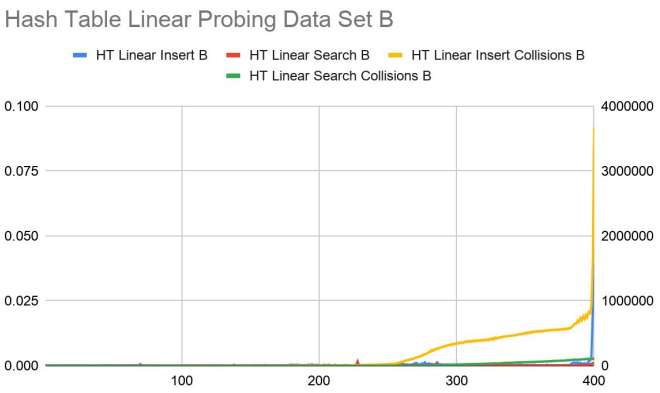
Figures 3 & 4: Average search and insertion times using a Binary Search Tree Data Structure for Data Set A and B respectively.



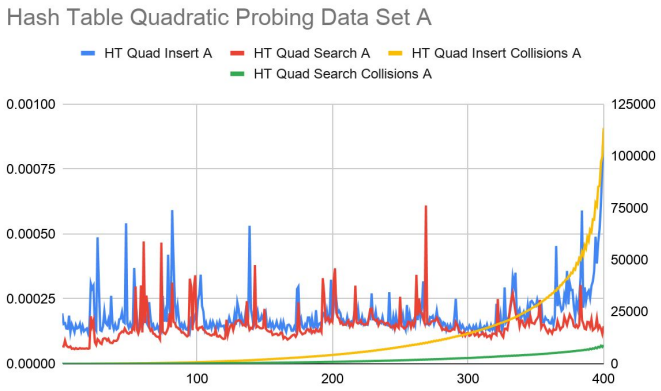


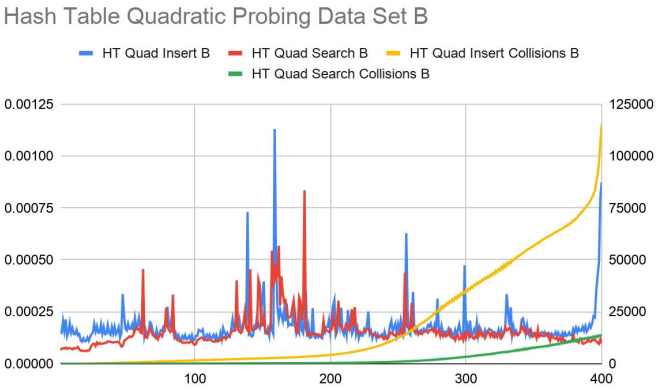
Figures 5 & 6: Average search and insertion times using a Hash Table Data Structure with Chaining Conflict Resolution for Data Set A and B respectively.





Figures 7 & 8: Average search and insertion times using a Hash Table Data Structure with Linear Probing Conflict Resolution for Data Set A and B respectively.





Figures 9 & 10: Average search and insertion times using a Hash table Data Structure with Quadratic Probing Conflict Resolution for Data Set A and B respectively.

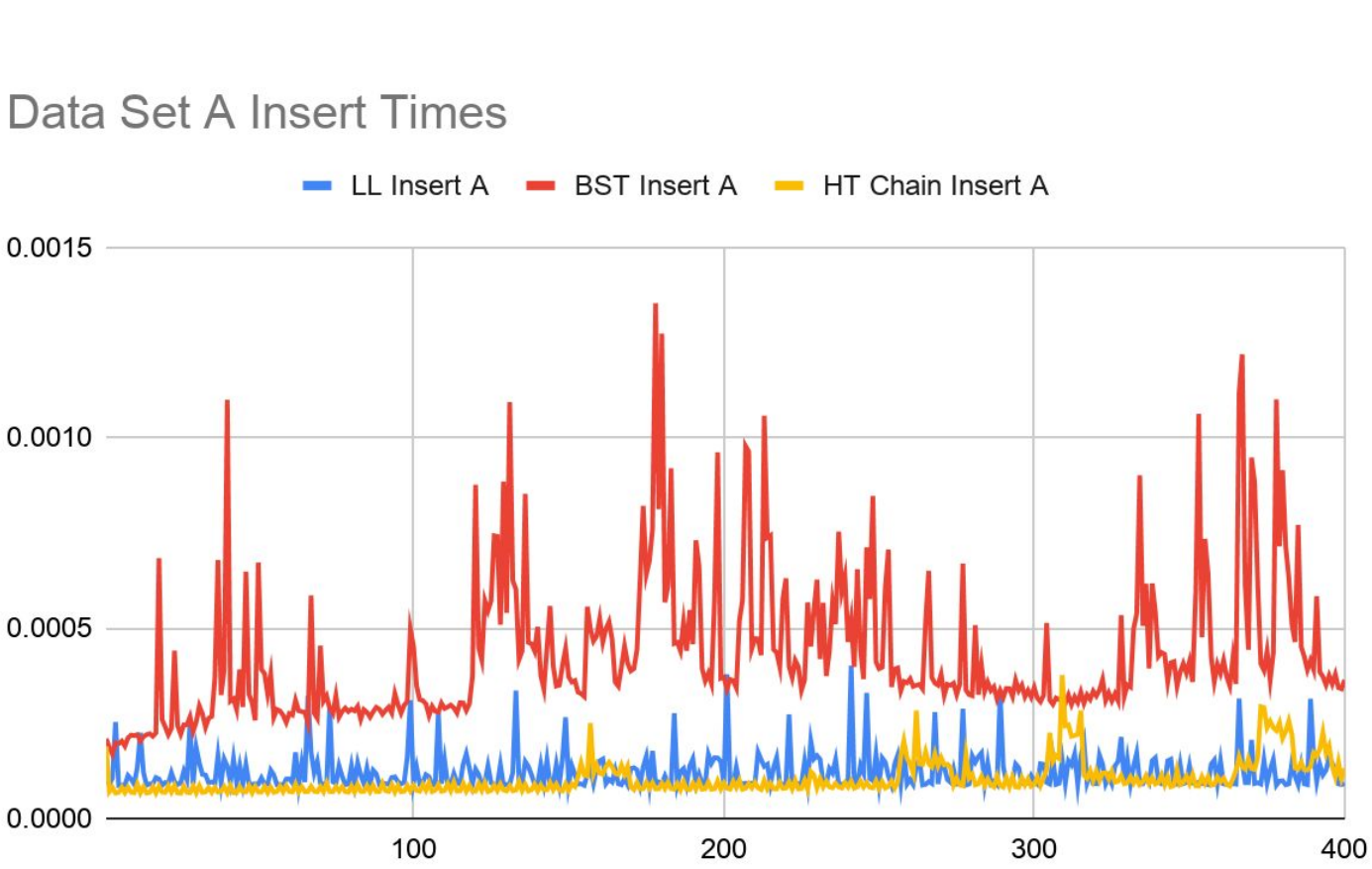


Figure 11: Average insertion times using Linked List, BST, and Hash Table: Chaining Data Structures for Data Set A.

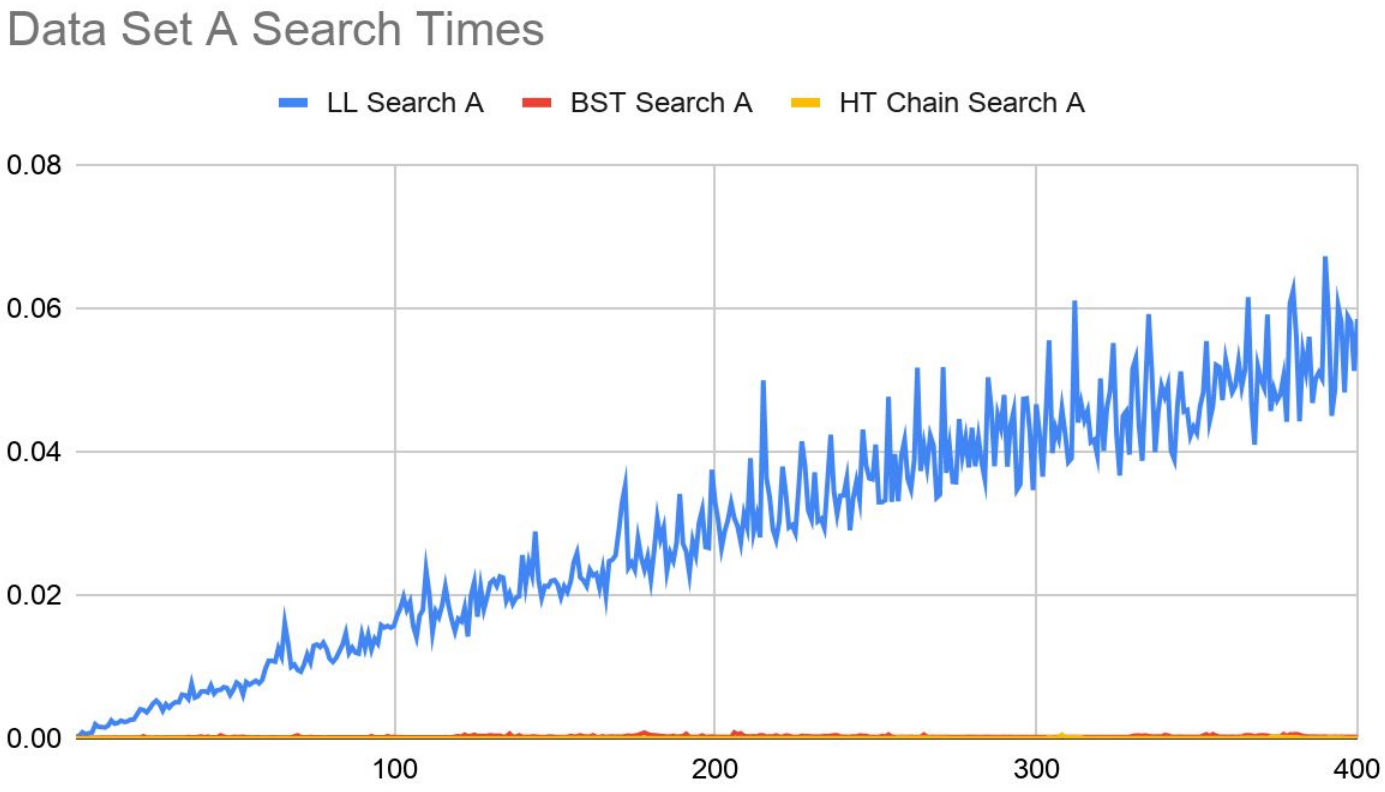


Figure 12: Average search times using Linked List, BST, and Hash Table: Chaining Data Structures for Data Set A.

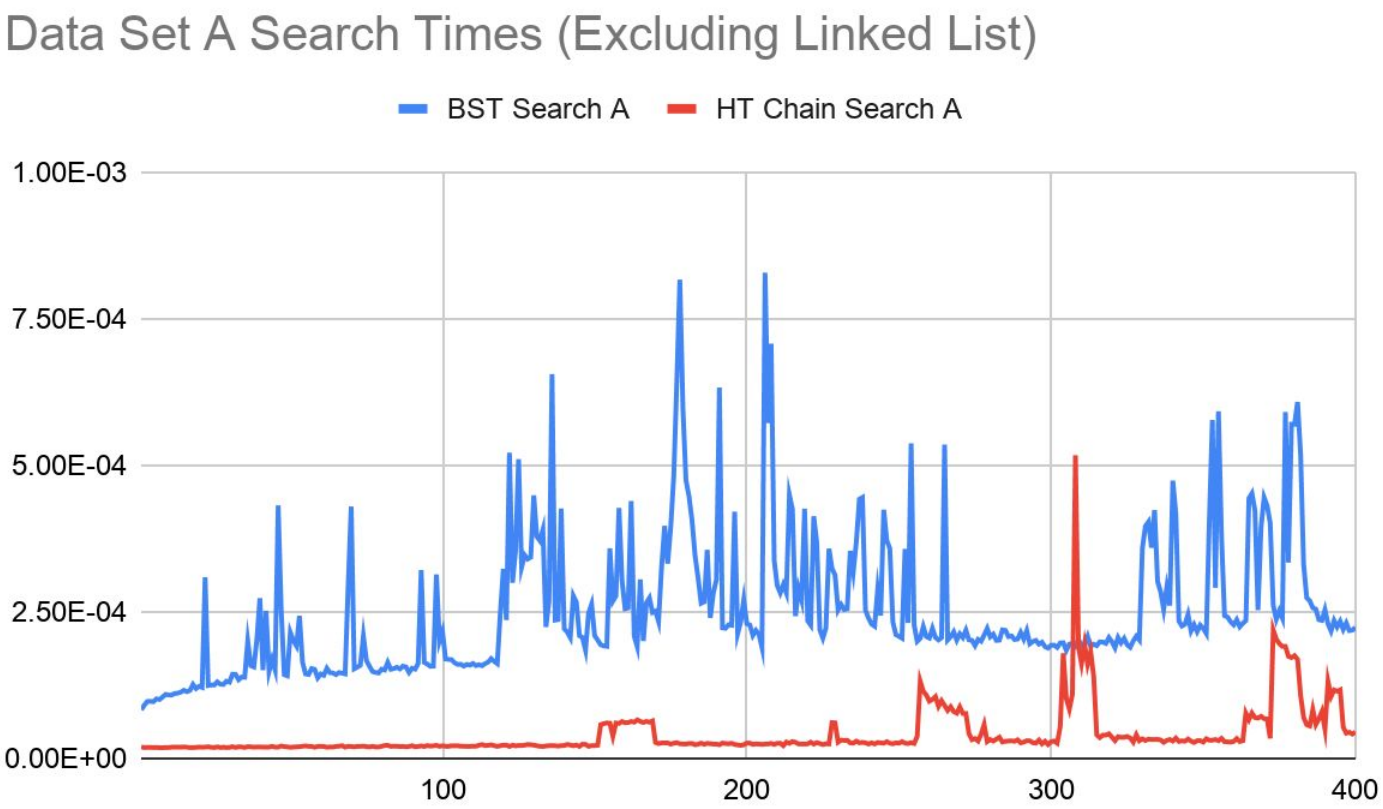


Figure 12.5: Average search times using BST and Hash Table: Chaining Data Structures for Data Set A. We chose to add this extra graph since on figure 12, you can't see which data structure performs better between a BST and the Hash Table with Chaining.

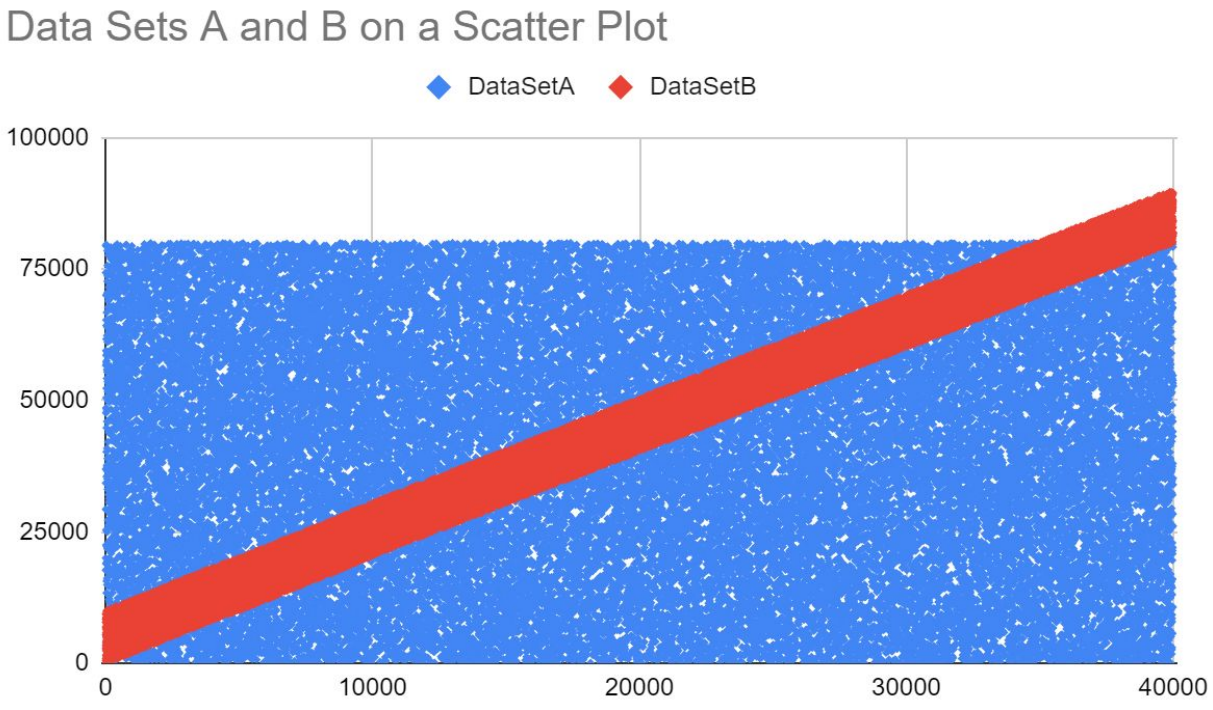


Figure 13: All of the data points from both data sets on a scatter plot.