Zachary Colleran

CSCI 2270

Asa Ashraf

7/19/2022

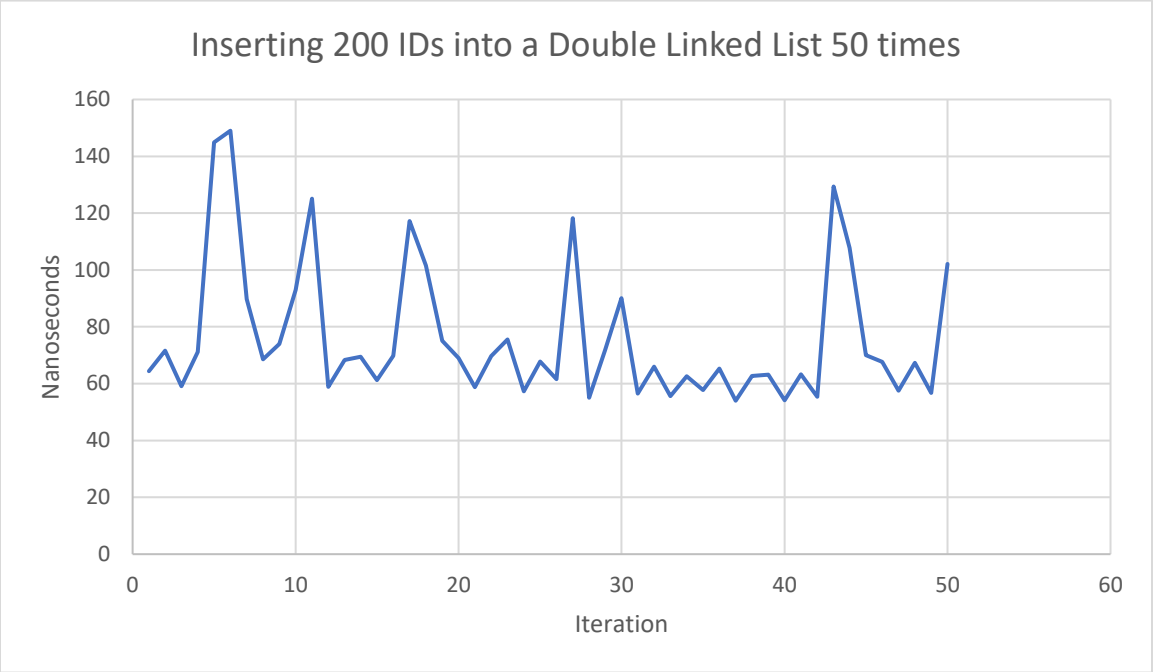
CSCI 2270 Project Report

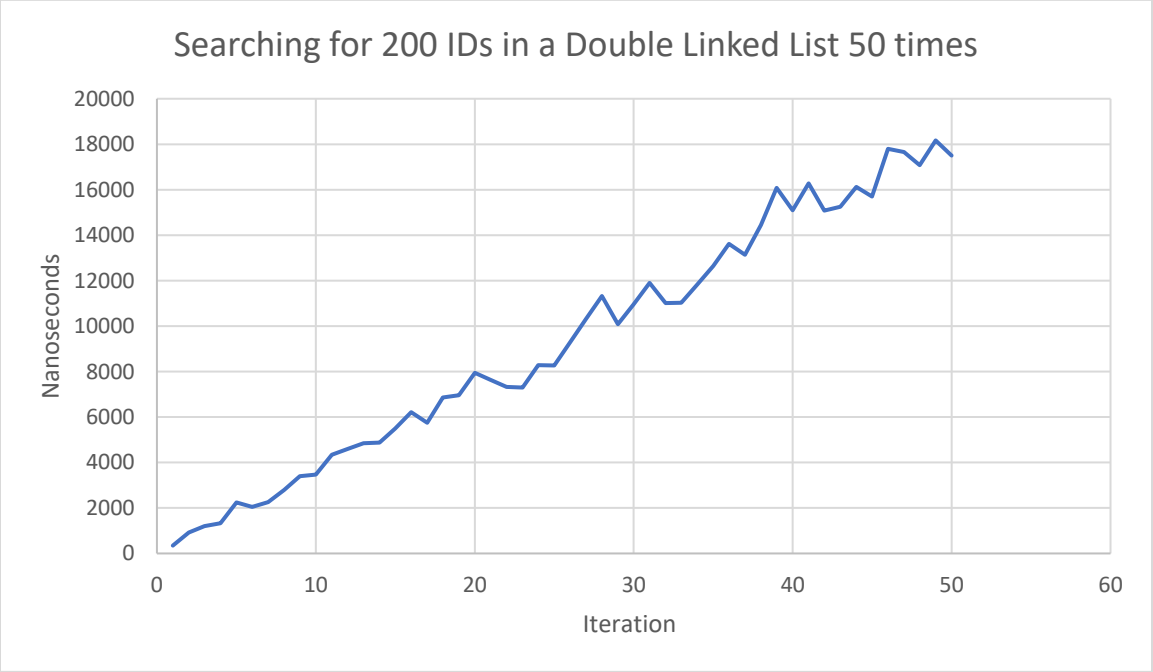
We were tasked to store a list of 10,000 IDs 3 different ways; Using a Double Linked List, Hash Table with BST Chaining, and Hash Table with Double Hashing. To check the time complexity, we calculated the time it takes to insert and search for IDs stored in each data structure. We did this by:

Inserting 200 IDs -> Searching 200 IDs -> Repeat (50 times)

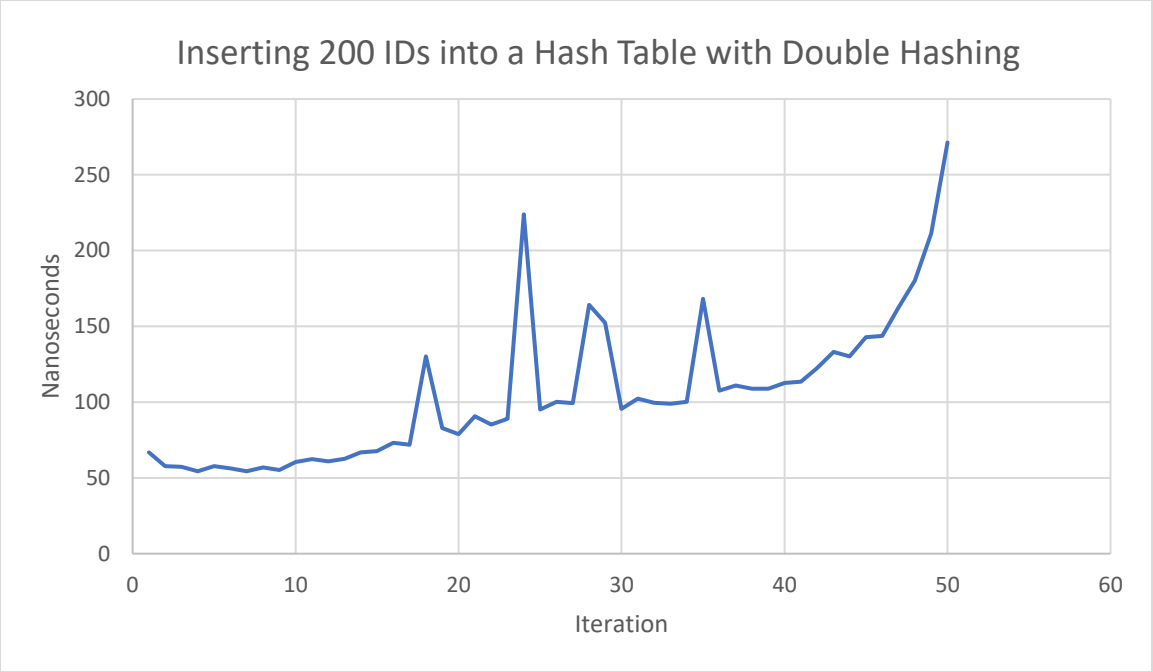
This way we could evaluate the time complexity of each data structure as the number of IDs stored increased. We also check the number of collisions the Hash Tables have as the # of IDs stored increases.

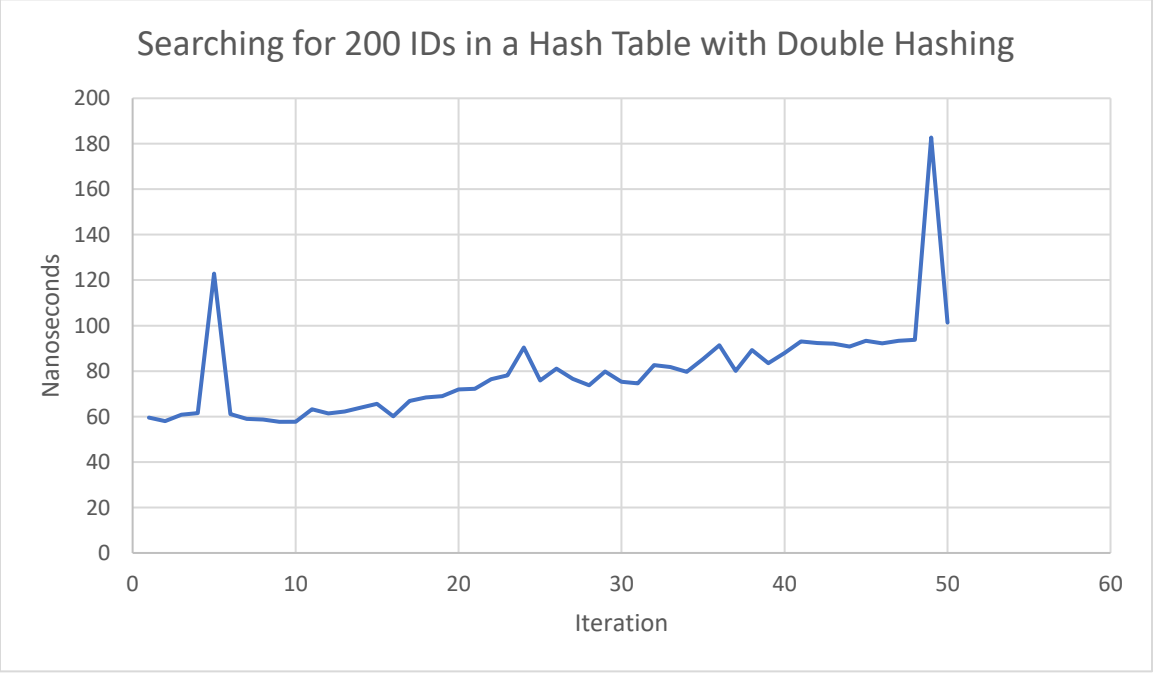
Double Linked List



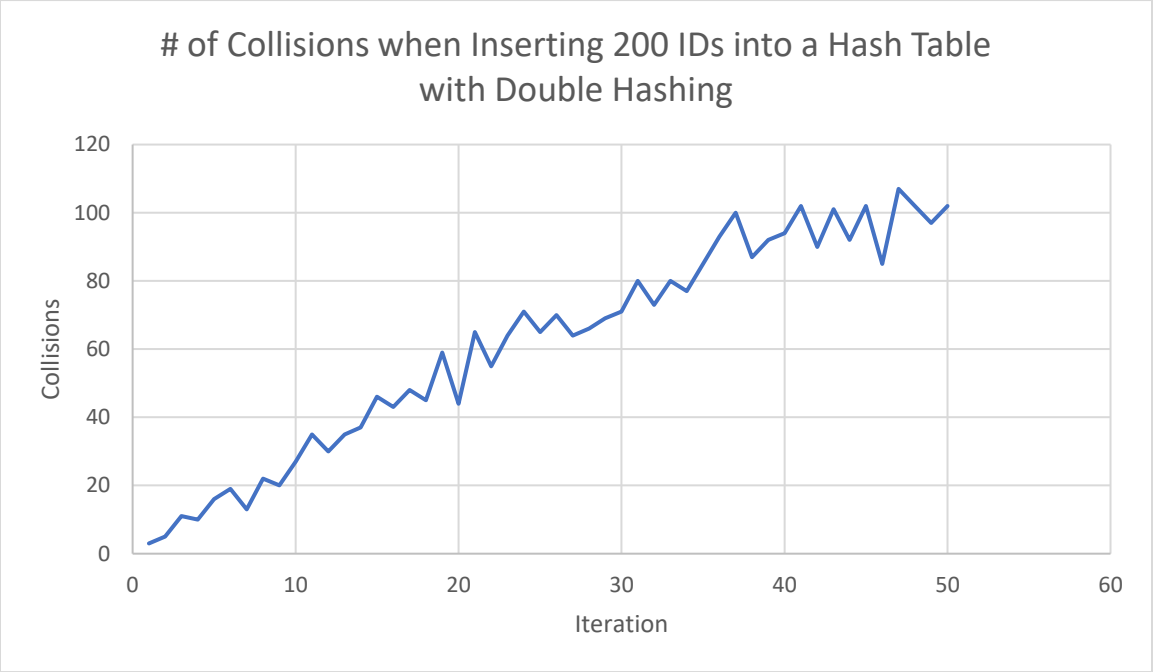


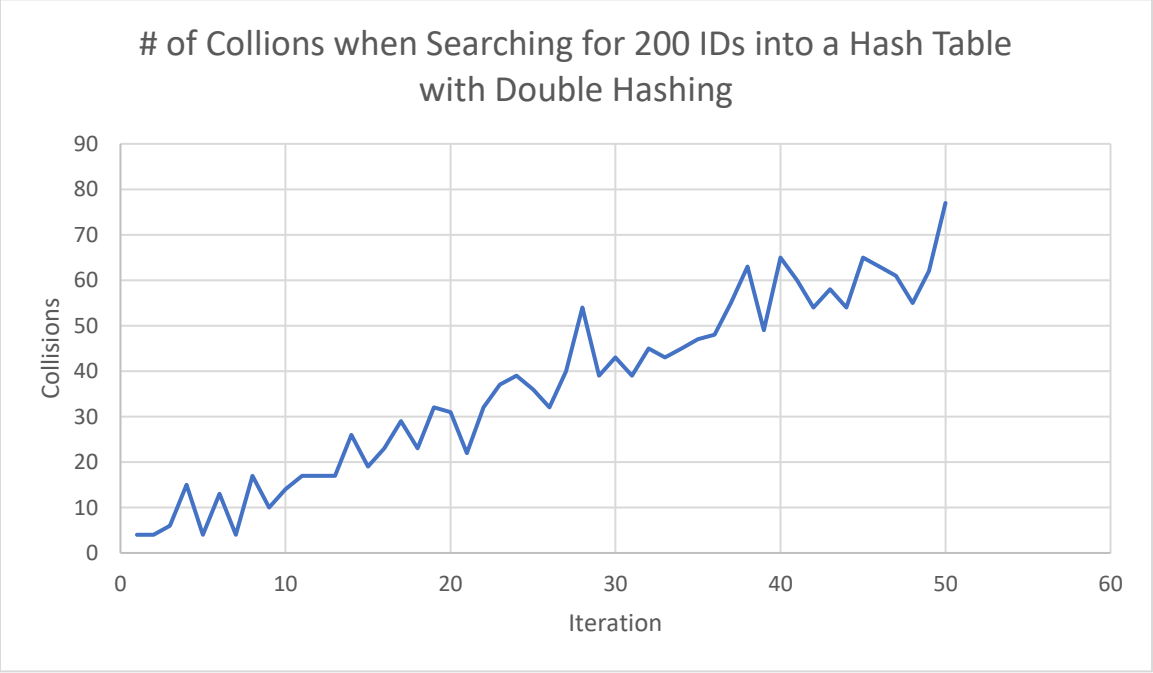
Hash Table with Double Hashing



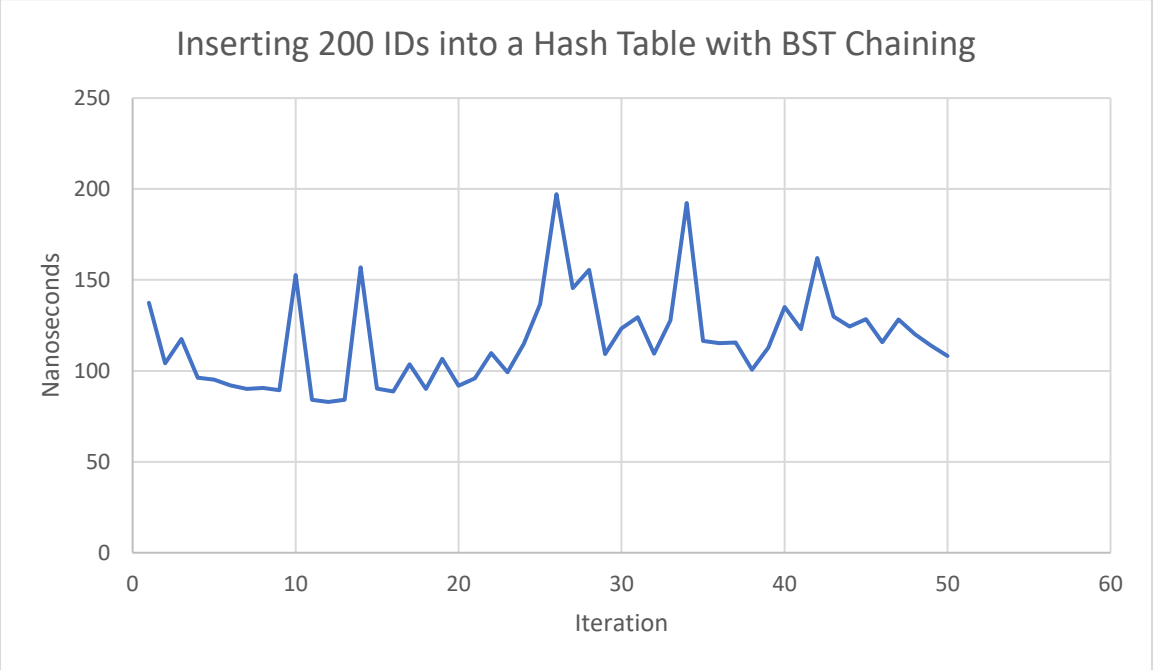


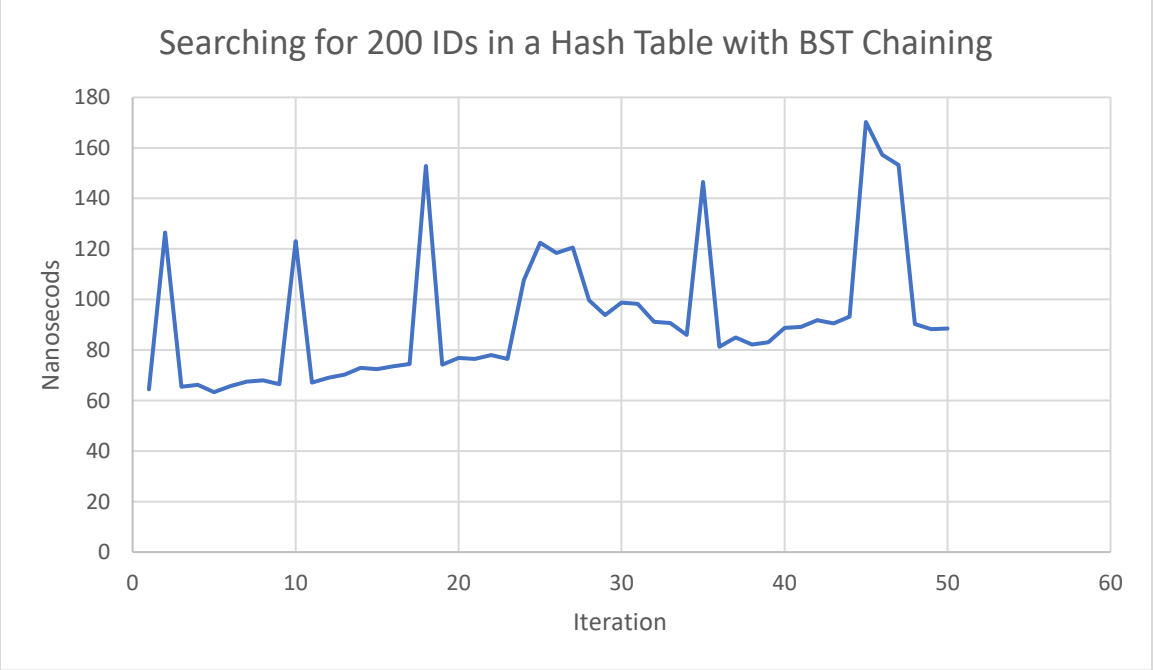
Double Hashing Collisions



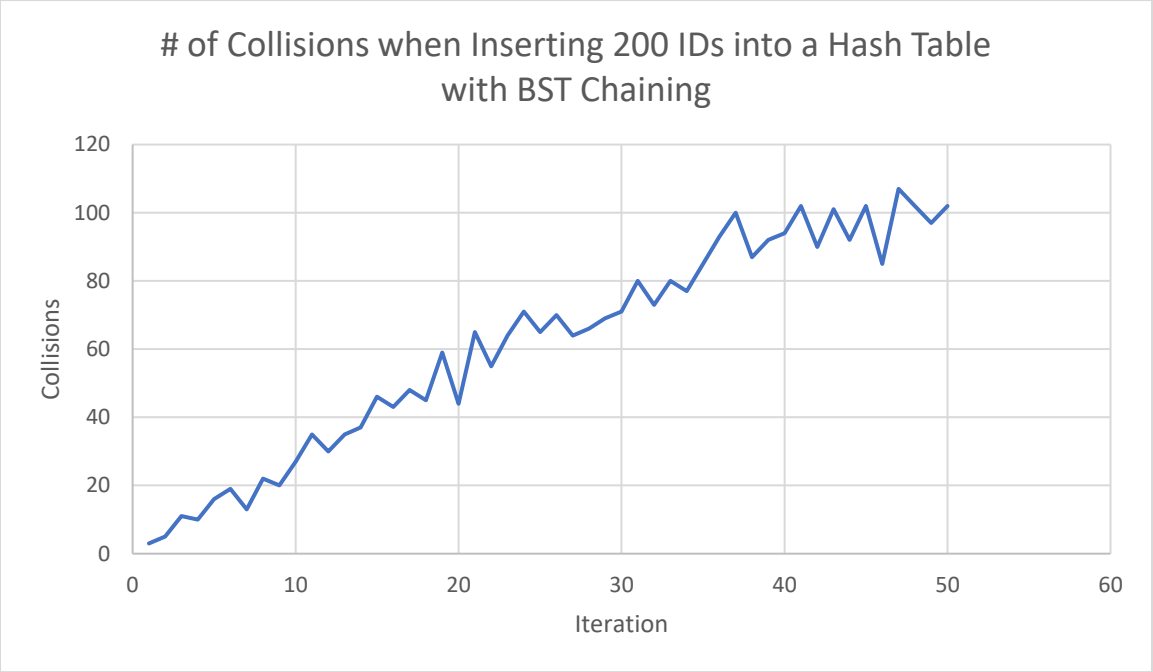


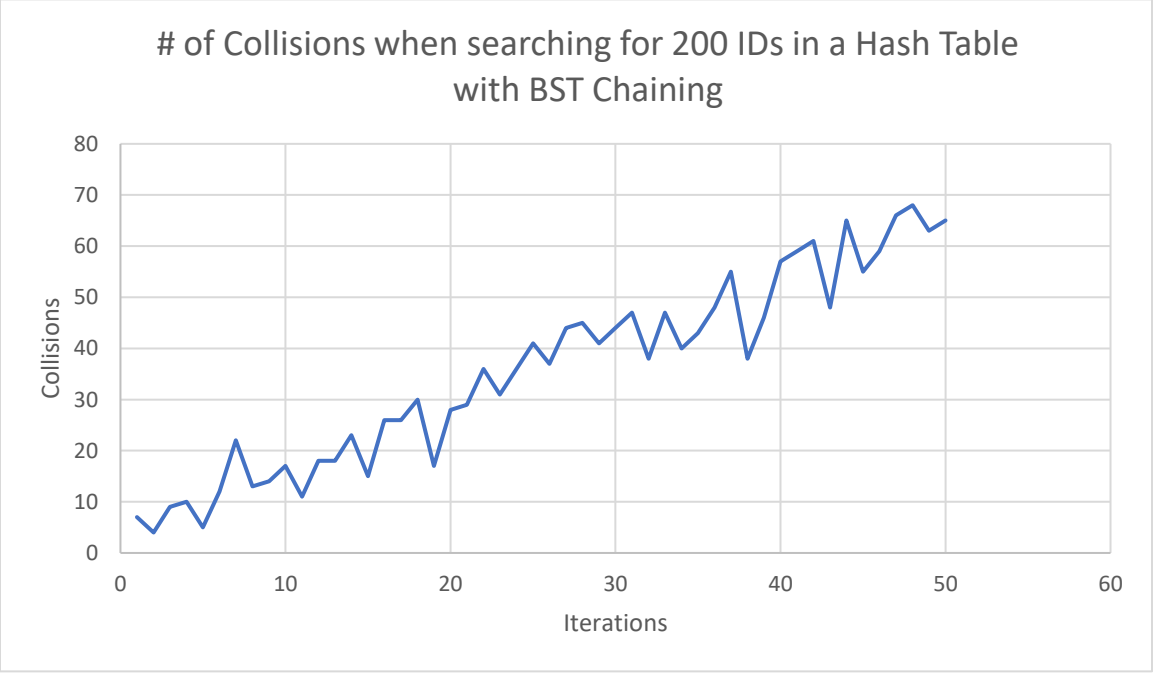
Hash Table with BST Chaining



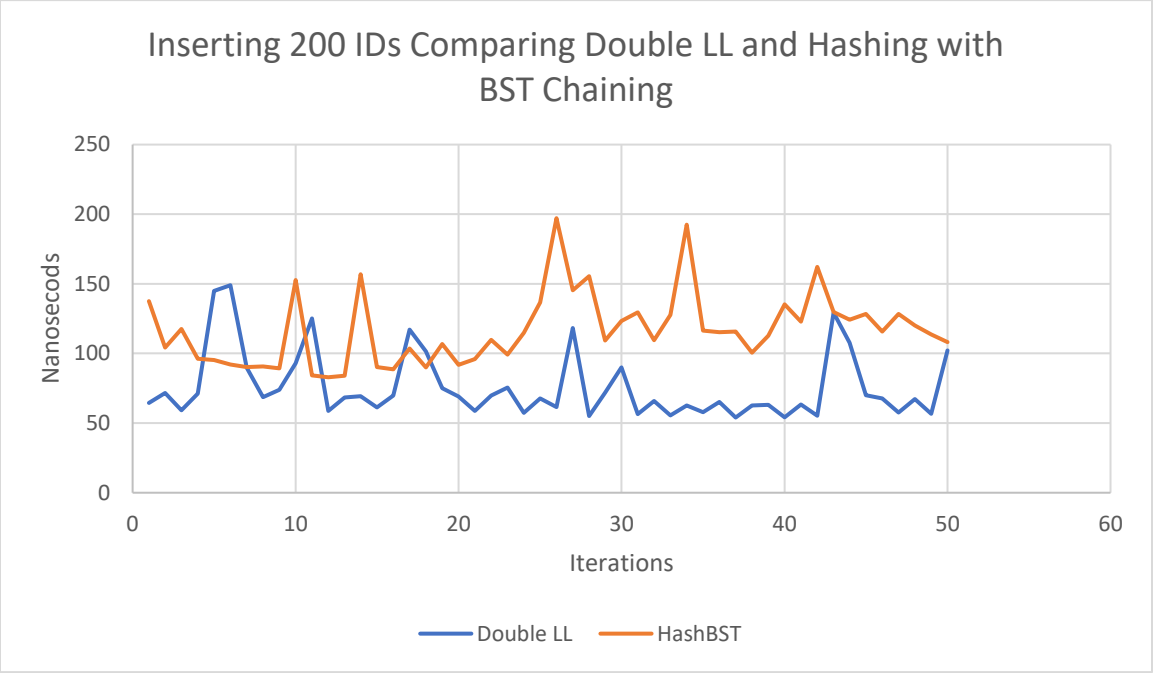


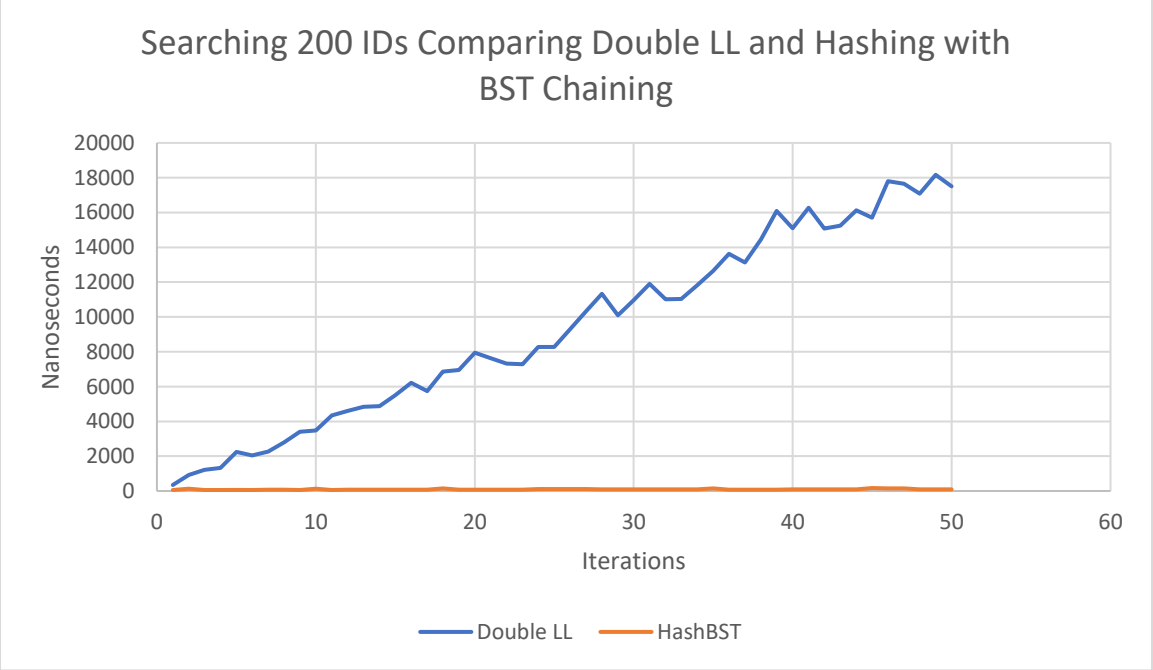
BST Chaining Collisions





Comparing Double Linked List and Hash Table with BST Chaining





Conclusion

From the comparison we can see that inserting into a Hash Table and a Double Linked List is very similar with the Double Linked List having a slight edge. Looking at the search comparison however, we can see the Hash Table with BST Chaining is immensely quicker than a Double Linked List. This conclusion makes sense because inserting into a Linked List and a Hash Table has the same time complexity O(1). On the other hand, searching in a Linked list has a time complexity of O(n) while a Hash Table is O(1).