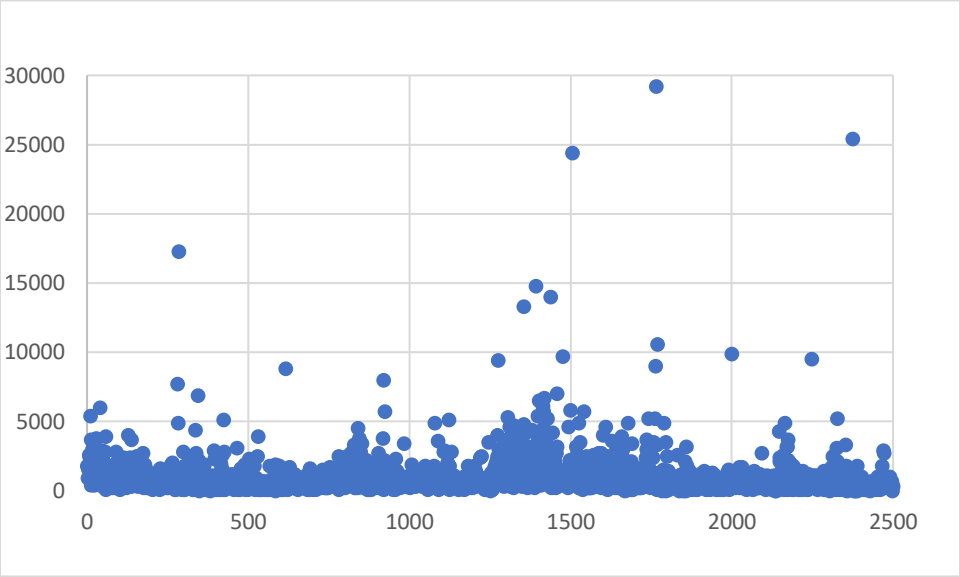
Binary Tree:

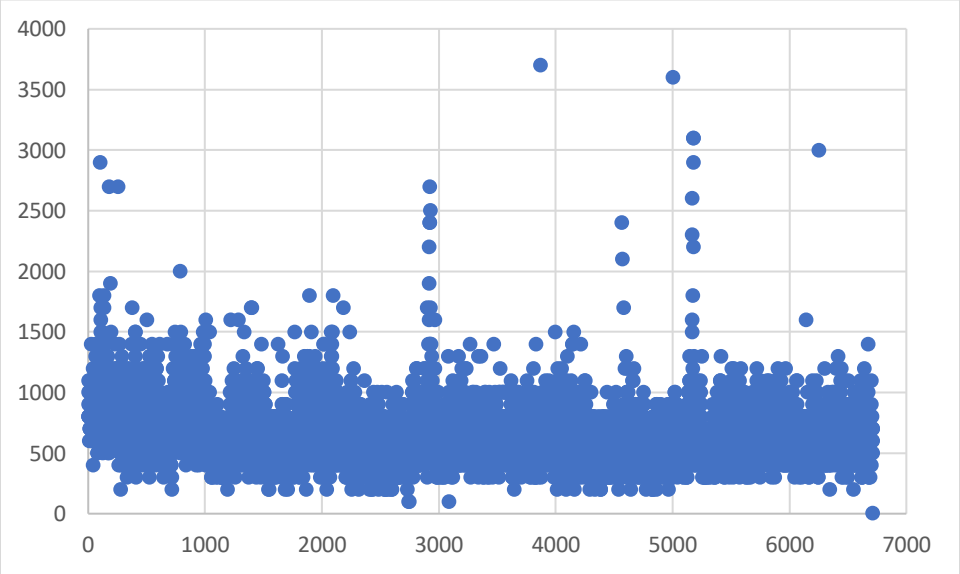
Insertion Plot:



Standard deviation: 22667.94 ns

Mean: 1640.198 ns

Search Plot:

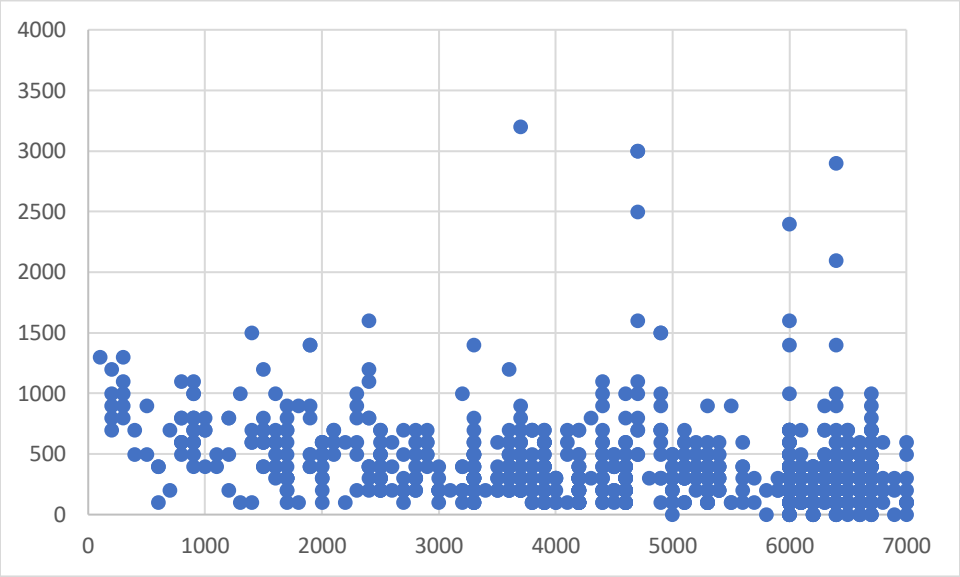


Standard deviation: 377.9691 ns

Mean: 676.2742 ns

Hash Table:

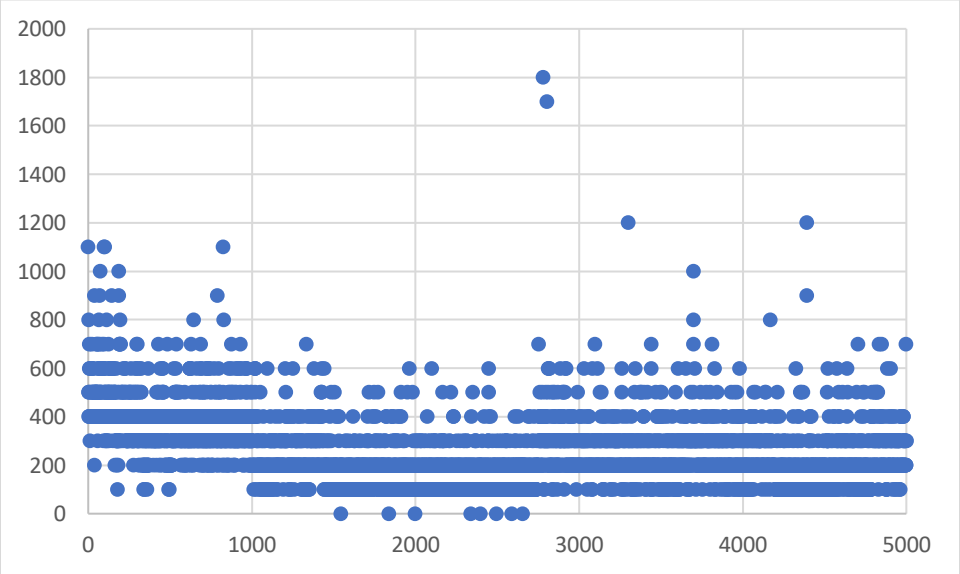
Insertion Plot:



Standard deviation: 1383.252 ns

Mean: 397.2332 ns

Search Plot:

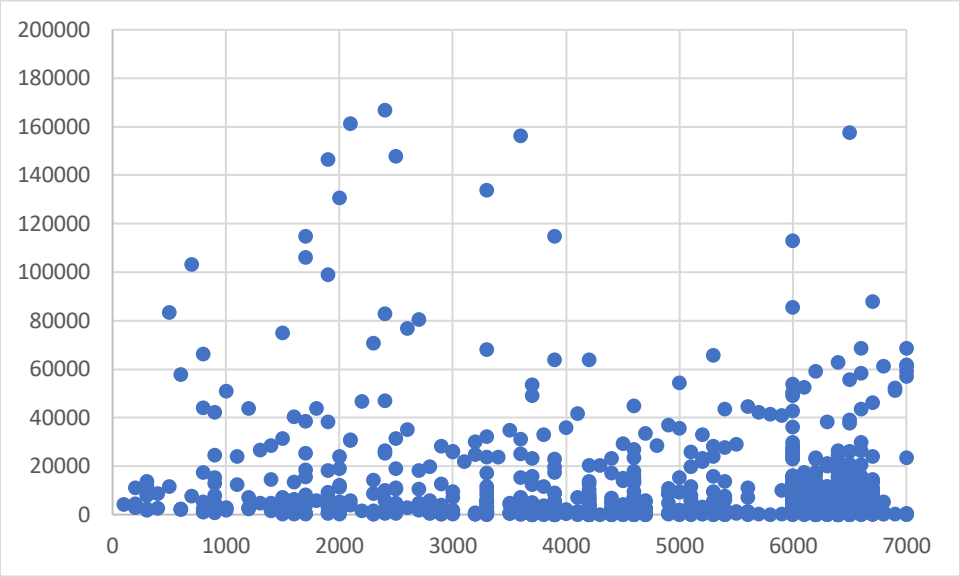


Standard deviation: 169.0952 ns

Mean: 260.9412 ns

Linked List:

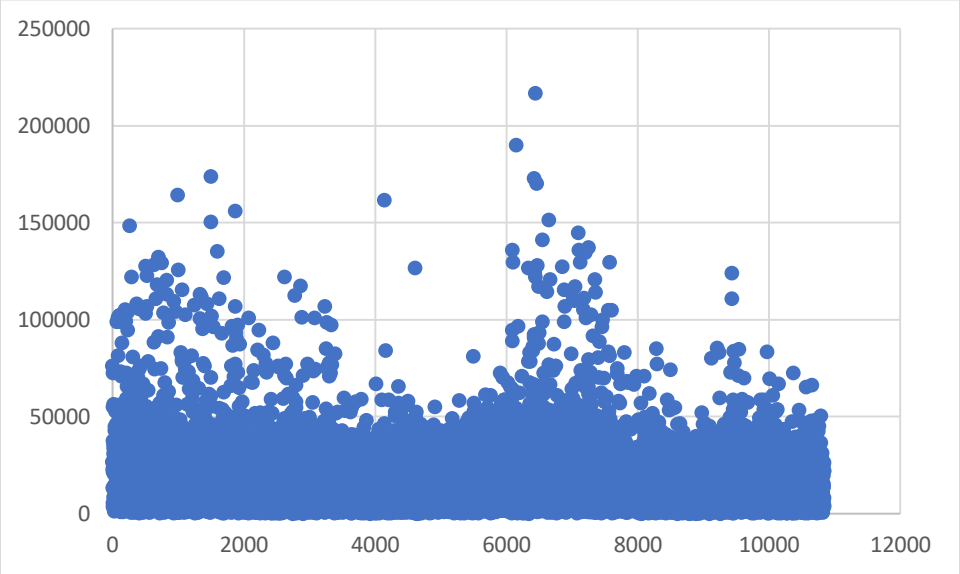
Insertion Plot:



Standard deviation: 26632.33 ns

Mean: 9400.791 ns

Search Plot:



Standard deviation: 286620.6 ns

Mean: 24137.11 ns

Observations are as below:

- Linked List takes up the most time for insertion and searching.
- There is a gradual increase in the time taken to insert as the number of elements in the Linked list increased. (Since the runtime for insertion is O(n))
- Hashtable insertion and searching is done in almost constant time. (Since the runtime for insertion and searching is O(1))
- Binary tree takes less time than linked list to insert and to search but more than hash table. Average Runtime for Binary tree is O(logn).