Comparison of Linked List & Binary Search Tree

# Time Execution for Linked List:

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **100**  (in microsecond) | **1000**  (in microsecond) | **10000**  (in microsecond) |
| **Add** | 10 | 8010 | 431158 |
| **Delete** | 10 | 7001 | 507085 |

# Time Execution for Binary Search Tree:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **100**  (in microsecond) | **1000**  (in microsecond) | **10000**  (in microsecond) | **100000**  (in microsecond) |
| **Add** | 150 | 2402 | 69129 | 559880 |
| **Delete** | 46 | 437 | 4930 | 19734 |

# Analysis of Results:

The Time to add in the linked is less than the time to add in the binary search tree. Similarly, Time to delete items from Linked List is less then the time to delete item from Binary Search Tree. So Linked list is much better than the Binary Search Tree is adding and deleting the Elements. In case of finding or searching the elements, the performance of binary tree is better than the performance of Linked List. So Linked List is better at adding and deleting Items while Binary Search Tree is better at Search or getting a particular Item.