# ***Analysis of Trees***

| **Tree Name** | LAZY AVL TREE | AVL TREE | BINARY SEARCH TREE |
|---|---|---|---|
| **Total Number of Recursive Calls to Insert** | 7362 | 7362 | 9680 |
| **Number of Nodes** | 565 | 565 | 565 |
| **Average Depth of Tree** | 7.4 | 7.4 | 10 |
| **Ratio of Average Depth to Log** | $\frac{7.4}{9.1} = 0.81$ | $\frac{7.4}{9.1} = 0.81$ | $\frac{10}{9.1} = 1.1$ |
| **Total Number of Successful Queries Before Deletion** | 420 | 420 | 420 |
| **Number of Recursive Calls to Contains Method Before Deletion** | 2965 | 2965 | 3801 |
| **Total Number of Successful Removal** | 210 | 156 | 156 |
| **Number of Recursive Calls to the Remove Method** | 1832 | 1706 | 2065 |
| **Number of Nodes After Deletion** | 565 | 409 | 409 |
| **Average Depth of Tree after Deletion** | 7.38 | 6.89 | 9.92 |
| **Ratios of Average Depth to Log after Deletion** | $\frac{7.38}{9.14} = 0.81$ | $\frac{6.89}{8.68} = 0.79$ | $\frac{9.92}{8.68} = 1.14$ |
| **Total Number of Successful Queries After Deletion** | 420 | 134 | 134 |
| **Number of Recursive Calls to Contains Method After Deletion** | 2965 | 3376 | 4021 |

- AVL and Lazy is better at inserting than Binary search tree and has a better average Depth of Tree
- AVL and Lazy is faster when using the contains search
- Lazy Deletion uses more space
- Surprisingly the AVL tree is faster at calling the remove function
- Expected that the AVL Tree to perform best of the three but deletes slower that Lazy AVL
- Expected that Lazy AVL Tree has faster deletion and after deleting fewer call to contains is need
- BinarySearchTree perform the worst of the three tree in all cases