



In the worst case scenario, the add or insertion method will have to add the element to the bottom of the longest branch of the tree. Therefore, big oh is $O(h)$ where h is the height of the tree. This is because the recursiveAdd would need to be called on each node till we reach the bottom of the tree where it is null and the element can be added there (following the path whether the element is smaller or larger than the current node value).

In this graph above which plots time (ps) vs height (# of nodes), there are a couple outliers here and there (2 ps at 13 nodes height), but generally the overall trend is to the right. This seems to be confirm the big oh that we have $O(h)$ in the worst case scenario. As we have more nodes in the height, generally the more time it takes to insert an element into the tree.