

Link to GitHub Repository:

<https://github.com/AlexBard122/Assignmnet5>

Note: We were unable to have our code finish executing due to running into stack overflow issues when creating the binary search trees with the additional updating of the left and right child counting fields. Therefore, there are no available task runtimes to provide for this assignment.

Task 1: Complexity order $O(n^2)$ due to every node n in a binary tree needing to be updated when a new node is added for a report n

Task 2: Complexity order $O(\log n)$ due to the method used in this task halving the number of possible nodes which it needs to check each time it iterates in its worst case scenario.

Task 3: Complexity order $O(\log n)$ due to the method used in this task halving the number of possible nodes which it needs to check each time it iterates in its worst case scenario.

The approach of using a recursive method to find the number of children rather than keeping that information in separate data fields is superior as the latter method increases both the space and runtime of code whereas a recursive method adds no space and requires no extra operations to update the number of children at each addition. This also greatly reduces the complexity of the code and reduces the chances for errors. Therefore, the best approach to use in a final product would be to use the recursive method and do away with the additional fields in the nodes entirely.