Uygur Tepe
105006877
tepe@uwindsor.ca

Data Structures and Algorithms Assignment 5 Analysis

insert Method:
The insert method implemented has an average case of O(log(n)) and worst case of O(n).
The average case is O(log(n)) because ideally each tree node will have 2 children so after iteration the current node goes either to the left or the right which halves the amount of possible nodes that could be the parent of the node that's to be inserted after each iteration. Therefore, as in this case there are 15 elements, on average in order to insert the newNode only 4 iterations are needed which is around log(15). So on average after each iteration there is half the amount of possible nodes which will be the parent node to the newNode.
The worst case is O(n) because if each inserted node is larger or smaller than its parent, then each node will only have one child. So if this happens then the tree will be heavily unbalanced and look more like a LinkedList as each node only has a right or left child, and traversing and inserting in a linkedList is O(n), thus the worst case search is O(n).

search Method:
The search method implemented has an average case of O(log(n)) and worst case of O(n).
The average case is O(log(n)) because ideally each tree node will have 2 children so after iteration the current node goes either to the left or the right which halves the amount of possible nodes that could be the element that is being searched for after each iteration. Therefore, as in this case there are 15 elements, on average in order to find if the entered value is in the tree only 4 iterations are needed which is around log(15). So on average after each iteration there is half the amount of possible nodes which contain the entered value.
The worst case is O(n) because if each inserted node is larger or smaller than its parent, then each node will only have one child. So if this happens then the tree will be heavily unbalanced and look more like a LinkedList as each node only has a right or left child, and traversing a linkedList is O(n), thus the worst case search is O(n).

remove Method:
The remove method implemented has an average case of O(log(n)) and worst case of O(n).
The average case is O(log(n)) because ideally each tree node will have 2 children so after iteration the current node goes either to the left or the right which halves the amount of possible nodes that could be the element that is being removed after each recursive call. Therefore, as in this case there are 15 elements, on average in order to find the element

Uygur Tepe
105006877
tepe@uwindsor.ca

to be removed in the tree only 4 iterations are needed which is around log(15). So on average after each iteration there is half the amount of possible nodes which contain the entered value for removal.

The worst case is O(n) because if each inserted node is larger or smaller than its parent, then each node will only have one child. So if this happens then the tree will be heavily unbalanced and look more like a LinkedList as each node only has a right or left child, and traversing and removing an element from a linkedList is O(n), thus the worst case removal is O(n).