Ads assignment 7
Jovan Shandro

Problem 7.2

We now do an asymptotic analysis of the algorithm we implemented for point b. The best case scenario is when the tree looks like a linked list itself, which means when every node has one child only, and in this case one of the 2 recursive call returns always null. So we will have n-1 recursive calls and since all other parts of the algorithm will only require constant time as the operator + will take only O(1) as one of the lists is always null, so in this case the complexity will be n*O(1) = O(n) where n is the number of nodes.
The worst case scenario is when the tree is perfectly balanced, which means that the number of nodes on the left is equal to (or differs by only 1 as n can be even or odd) to those on the right. In this case the recurrence resembles that of the merge sort it is something like this $T(n) = 2T(n/2) + O(n)$, the O(n) in the end is from the summation of the lists which is done in linear time. So the asymptotic time complexity is O(nlgn) in worst case scenario.