CS5800: Algorithms Spring 2018
Assignment 5.4

Saptaparna Das

March 16, 2018

Both implementations implement the same data structure, and the same operations on it. The routine binary search tree is easier to understand and implement since we get to access the left or right nodes from parent. The implementations are pretty straight forward in most of the cases.

The composite one might seem a little complicated at first since there is no standard left/right children design. But this is more functional way of doing things. The emptynode has been seperated from nonempty one so its loosely coupled and more efficient. Later if the logic of nonempty node changes, we won't need to touch the emptynode operations. All the operations are done in their respective relevant classes. E.g. Node level operation are happening in Node classes whereas operations related to complete tree are happening in the tree implementation.

In routine binary search implementation all the node level operations are also possible to be done on tree level which might be easy to do but violates class semantics. Moreover, the composite one seems more modularized also. Even though there is not much of difference in time or space complexity of both the implementations, the composite one seems to have a better design to me which easily can be reused without breaking existing functionalitites