

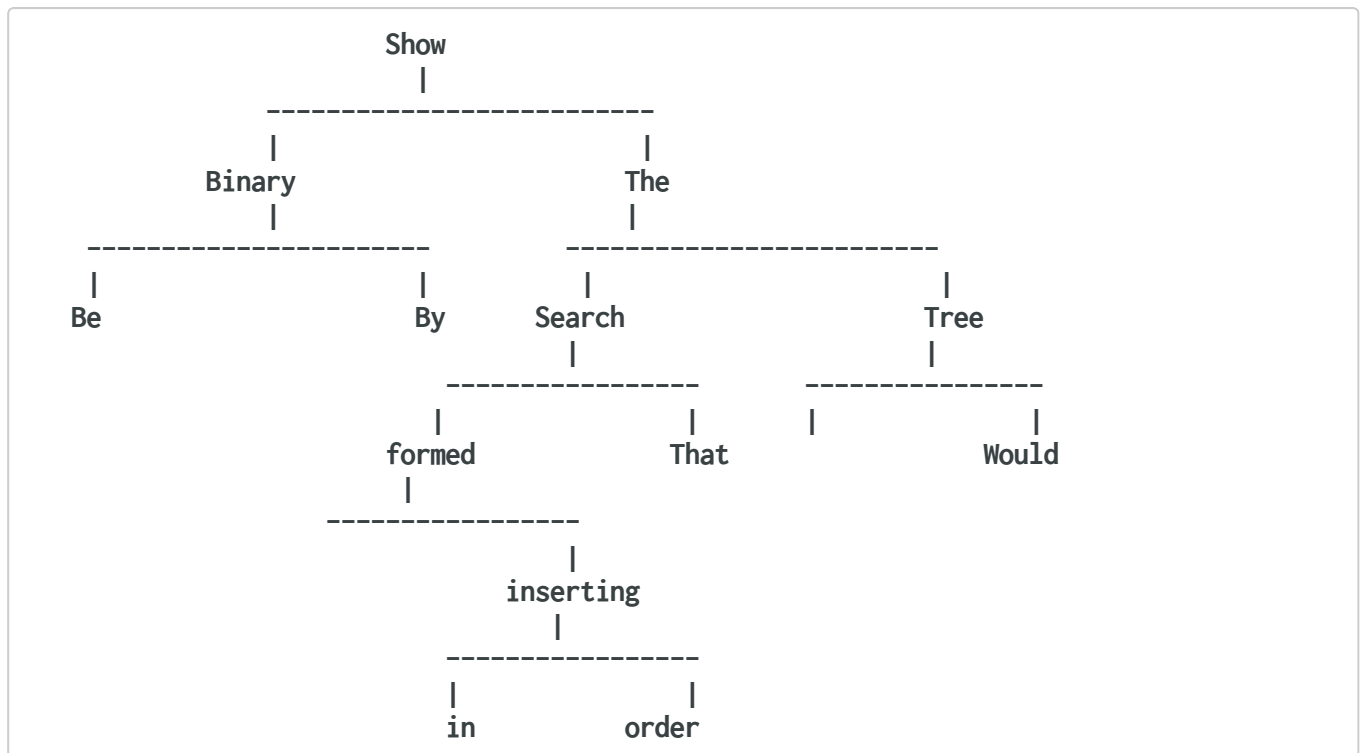
CS2010 Tutorial 3 Homework

Feb 4, 2014

TONG Haowen Joel

A0108165J

Question 1



Question 2

Part A

In a binary search tree, keys which are smaller are assigned on the next available left node, while keys which are larger are assigned on the right node.

If no slots are available, then the keys propagate in that order till an empty leaf slot is found.

In the case of **father**, the keys will propagate in that sequence till farmer is reached. Since according to ASCII encoding weightage, **T** is greater than **R**, therefore **Father** will be inserted as the right child of **Farmer**.

This will change height of tree, as farmer is already the lowest leaf node of the BST.

Part B

In the case of `knapsack`, the height of BST remains the same as it is not the lowest leaf node of BST.

Question 3

If there are no children (i.e. leaf node), just delete node.

If there is one child, delete node and move the rest of nodes up.

If there are two children, delete node and replace that node with the minimum node of the right child (of old tree).

Dog

Dog has 2 children, therefore delete `dog`, and move `farmer` node to `dog` position.

and

`and` has 2 children, therefore delete `and` and move `ate` to replace `and` position.

Rat

`rat` has 2 children, therefore delete `rat` and replace with `shaven`.

Question 4

If alternate replacement is used, i.e. just simply inserting any value into tree, in the long run, the tree may be skewed. This leads to degradation of performance to $O(n)$.

Question 5

If there is no children, then that means that the node simply has to be deleted without need for bubbling up nodes. Therefore, checking for the nonexistence of both children is not necessary.