Eclipse, Kliezl P. Exercise 4.5 – 8

Q:

a. Outline an algorithm for deleting a key from a binary search tree. Would you classify this algorithm as a variable-size-decrease algorithm?
b. What is the time efficiency class of your algorithm in the worst case?

A:

a. When deleting a key from a binary search tree, three possibilities can happen:

    Case 1: *The key's node is a leaf*

    What to do: **Remove node from tree**

    Case 2: *The key's node has one child*

    What to do: **Copy the child node, delete the key's node, replace it with the child node**

    Case 3: *The key's node has two children*

    What to do: **Find either the minimum in the right subtree of the key's node or the maximum in the left subtree, *M*, copy *M* to the key's node, and delete the duplicate *M* from the subtree.**

    Note: deleting the duplicate *M* may perform the 3rd case in every iteration until it reaches either the 1st or 2nd case.

    With this, deleting a key from a binary search tree can be classified as a variable-size-decrease algorithm since the size-reduction pattern varies on the number of children of each node, and the key of each node.

b. The time efficiency of the algorithm in the worst case is O(h) where h is the height of the tree.