# Michael Chillemi

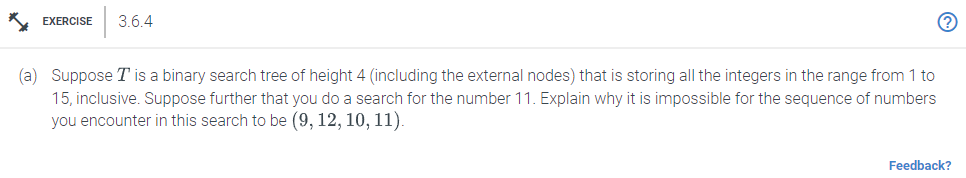
# 06/02/2023

# CS 590 - Algorithms

# M3.B1: Module 3 Binary Search Trees Reinforcement Exercises

# 

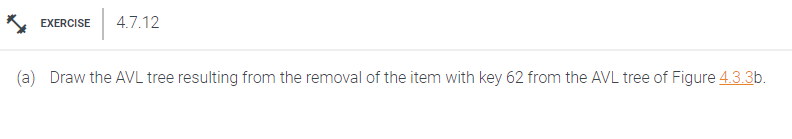
Problem 3.6.4



Answer:

The series of numbers you come across in this search cannot be (9,12,10,11). In a Binary Search Tree, every node in the current node's left sub-tree is smaller than the current node, and every node in the current node's right sub-tree is larger than the current node. The nodes visited during an element search in the tree either have elements in decreasing or rising order, not both.

Problem 4.7.12



Answer:

In an AVL tree, a self-balancing binary search tree, every node's left and right subtree heights differ from one another by no more than one. Each node in an AVL tree stores a key-value pair and has two left and right child nodes. In contrast to the keys in the node, the keys in the right subtree are greater than the keys in the left subtree. Finding a key in the tree is straightforward since we can compare the key to the key of the node and choose whether to search the left or right subtree. First you would add 52 since 52 is less than the root node it will go to the left sub tree. Since 52 is greater than 44 then it will look at that right sub-tree. Then you would check and see 52 is less than 54 and 54 has no children so 52 will be inserted in the left subtree of 54. After that you will apply a single rotation left. Next you will remove 62 as the root node and replace it with 54. If the node to be deleted has two children, the largest node in the left subtree can be located. The highest value of the left subtree is copied to the node that will be destroyed. Which will give you the final AVL Tree shown below.

AVL Tree:

