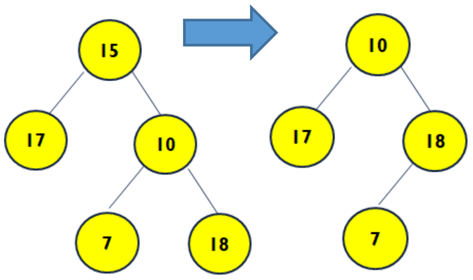
INTRODUCTION TO ALGORITHMS (CELEN086)
EXTRA PRACTICE PROBLEMS (5)

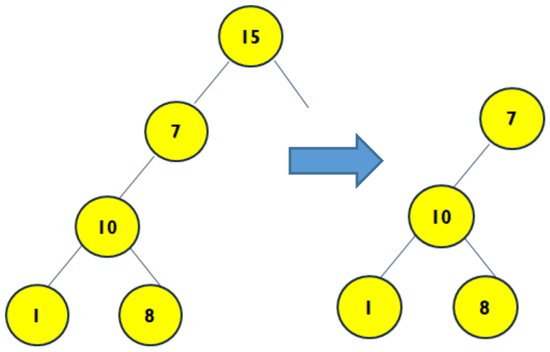
TOPIC: ***Binary Search Trees***

Note: ALL algorithms must come with proper headings. Also, trace your algorithms!!

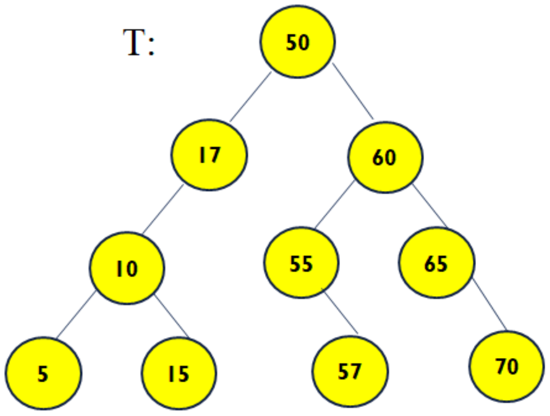
1. Write a recursive algorithm called **`L2BST(L)`** that takes a sorted list of numbers and makes a minimal balanced binary search tree. You can use **`length()`**,**`getNth()`** and **`cut()`** functions that we have written before in class. Trace your algorithm for **`L=[2,4,6,8,10]`**.

2. Write a recursive algorithm called **`delRoot(binT)`** that takes a binary tree and deletes the root and returns a new binary tree whose root is the root of the right subtree. If the right subtree is empty then the algorithm should return the left subtree. Trace your algorithm for the following examples:





3. Write a recursive algorithm called **`belX(x,bST)`** that takes a binary search tree and a value x and counts the number of nodes whose values are less than x. You can call the **`treeSize()`** algorithm from Seminar 7. Then consider the following BST:



Trace your algorithm for: **`belX(20,T)`** and **`belX(58,T)`**

4. Consider the BST given in Question 3.
   i. Apply NLR, LNR and LRN traversal schemes to this tree.
   ii. Insert node values 0, 8, 18, 25, 66 and 80 to this tree. Redraw the BST with the new nodes
   iii. Apply the LNR scheme to the tree you obtained in (ii) above to obtain a sorted list.
   iv. Convert the list you obtained in (iii) above to a minimal balanced BST. Is it different from the tree you drew in part (ii)?