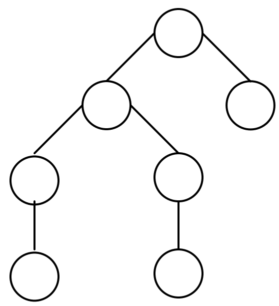
CSE 373 – Data Structures and Algorithms
Autumn 2003. Dry assignment #3
Due date: 11/7/03.

1. (*30 pts.*) Given a binary tree, let l_i denote the number of leaves at depth i. For example, in the following tree $l_0=0$, $l_1=1$, $l_2=0$, $l_3=2$. Also, let h be the height of the tree.



1a. Prove that $\sum_{i=0}^{h} l_i \cdot 2^{-i} \leq 1$. Hint: use induction on h.

1b. When does $\sum_{i=0}^{h} l_i \cdot 2^{-i} = 1$ hold? Give a necessary and sufficient condition and explain.

2. (*30 pts.*) A point in the 2-dimentional plane is given by its coordinates (x,y).

Suggest (and explain in words) a data structure that supports the following operations in the specified time complexity. Note: n is the number of points at the time the operation is performed.

insert(x,y) – adds a new point whose coordinates are (x,y). Time complexity: O(log n).

delete(x,y) – deletes the point whose coordinates are (x,y). Time complexity: O(log n).

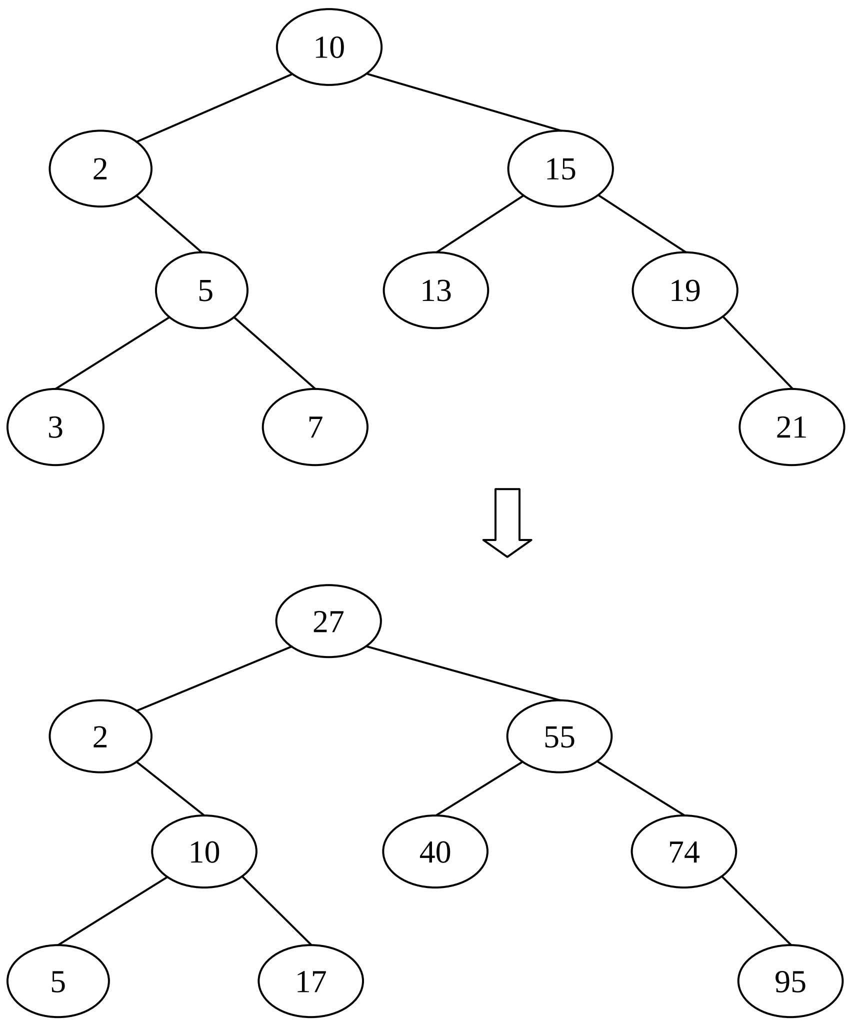
line(m) – prints out all the points (x_i,y_i) for which $x_i+y_i=m$. (These are all the points that are along the line x+y=m.) Time complexity: O(log n + k), where k is the number of points in the output.

Explain why these are the time complexities of the given operations on your chosen data structure. Hint: A single balanced search tree is not enough.

3. (*25 pts.*) Given a binary search tree TI (not necessarily balanced), suggest an algorithm that creates an additional binary tree TO of the exact same shape (see the example below) with the following property: Let k be the key of a node in TI, then the key of the node in the same position in TO is the sum of the keys from TI that are less than or equal to k.

You need to obey the following constraints: The input tree must not be changed. Your function has to allocate space for the new tree. Your solution should include pseudocode as well as an explanation in words. The time and space complexities of your solution should both be O(n) – be sure to explain why they are so.

Example: An input tree and the constructed output tree.



4. (*15 pts.*) Is there an AVL tree of height 4 that has exactly 11 nodes? If yes, draw such a tree and explain why it is an AVL tree; if not, explain why.