CSE326 — Data Structures, Winter 2004
Dry Assignment #3 Solutions

1. ℓ_i denotes the number of leaves at depth i in a binary tree of height h. We examine the inequality

$$\sum_{i=0}^{h} \ell_i \cdot 2^{-i} \leq 1$$

a. The inequality can be proven by induction on the height h of the tree. If $h = 0$, then $\ell_0 = 1$ and

$$\sum_{i=0}^{0} \ell_i 2^{-i} = \ell_0 2^0 = 1,$$

so our base case holds. Now assume the statement is true for all $h \leq n$, and consider a tree T of height $n + 1$ rooted at r. Let A and B be subtrees rooted at the children of r, and let ℓ_i^A and ℓ_i^B be the number of leaves in A or B of height i, measured from the root of A or B. Then for all $i > 0$, $\ell_i = \ell_{i-1}^A + \ell_{i-1}^B$, because a leaf of T at height i is a leaf of either A or B of height $i - 1$ in that tree. Furthermore, because the height of T is at least 1, $\ell_0 = 0$. So we have

$$\begin{aligned}
\sum_{i=0}^{h} \ell_i \cdot 2^{-i} &= \sum_{i=1}^{h} (\ell_{i-1}^A + \ell_{i-1}^B) \cdot 2^{-i} \\
&= \left(\sum_{i=1}^{h} \ell_{i-1}^A \cdot 2^{-i} \right) + \left(\sum_{i=1}^{h} \ell_{i-1}^B \cdot 2^{-i} \right) \\
&= \left(\sum_{i=0}^{h-1} \ell_i^A \cdot 2^{-i-1} \right) + \left(\sum_{i=0}^{h} \ell_i^B \cdot 2^{-i-1} \right) \\
&= \frac{1}{2} \left[\left(\sum_{i=0}^{h-1} \ell_i^A \cdot 2^{-i} \right) + \left(\sum_{i=0}^{h} \ell_i^B \cdot 2^{-i} \right) \right].
\end{aligned}$$

Now we can apply the induction hypothesis to the two inner sums to get

$$\frac{1}{2} \left[\left(\sum_{i=0}^{h-1} \ell_i^A \cdot 2^{-i} \right) + \left(\sum_{i=0}^{h} \ell_i^B \cdot 2^{-i} \right) \right] \leq \frac{1}{2} (1 + 1) = 1,$$

which proves the inequality.

b. We note that if a tree is empty, the sum is equal to zero. In this case one of the summands where we apply the induction hypothesis will be zero, and our inequality will not be tight. Conversely, if neither child is empty, they contribute to the sum, and if their sums are tight, the final sum will be tight.

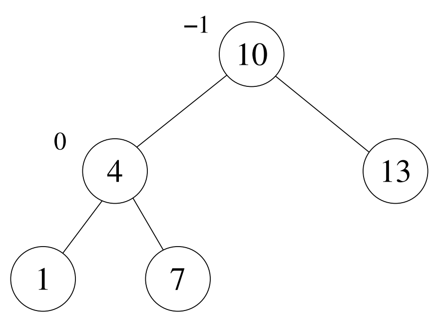
Thus a necessary and sufficient condition for the inequality to be tight is that each node is either a leaf, or has two children.

2. I've written the balance, $\mathrm{Height}(\text{right child}) - \mathrm{Height}(\text{left child})$, next to each internal node in the pictures.

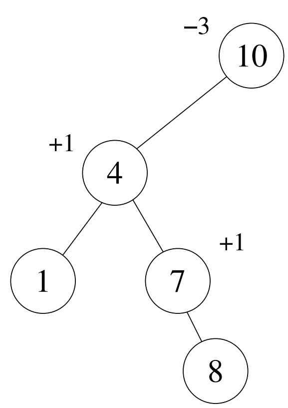
a.

1. No. An node is always inserted as a leaf.

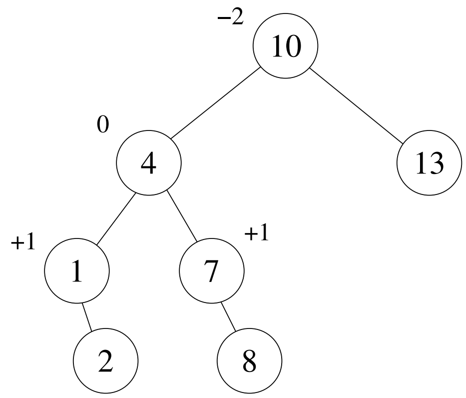
2. Yes.



3. No. If 13 were inserted, the tree prior to the insert would not be a valid AVL tree, as evidenced below.



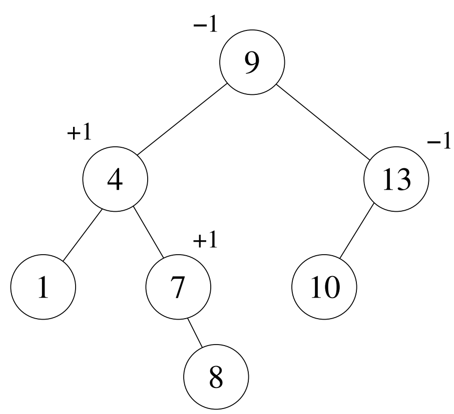
4. No. If 2 were deleted, the tree prior to the insert would not be an AVL tree. Even if 2 were an internal node, it would be on the left of the root.



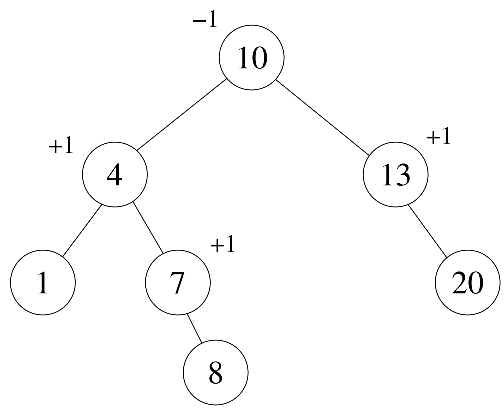
5. No, the root would be unbalanced just as above. (Other nodes would also be unbalanced, but one reason is enough).

6. Same as 4.

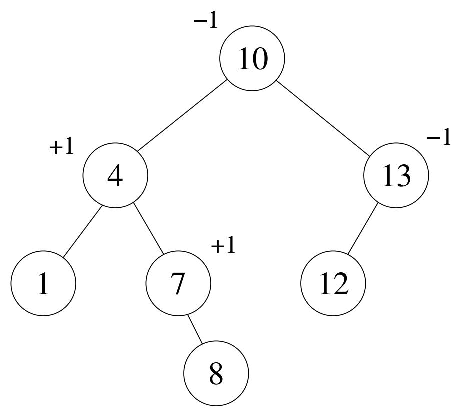
7. Yes. Note that 9 could have been the root of the tree, and we could be replacing deleted nodes by their inorder successors.



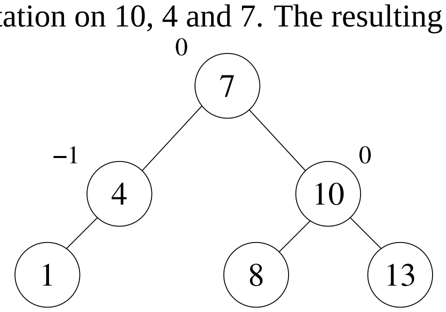
8. Yes.



9. Yes.

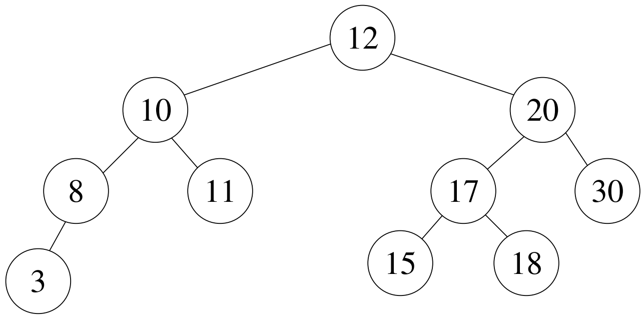


b. Assume 8 was just inserted. The root became unbalanced, so we do an inside rotation on 10, 4 and 7. The resulting tree is below.



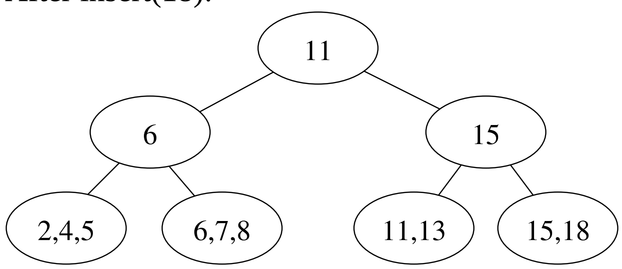
3.

a.

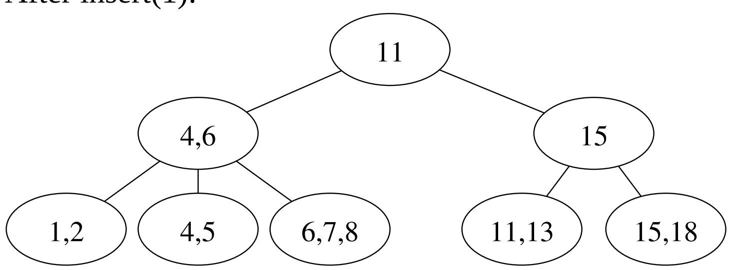


b.

After insert(18):



After insert(1):



After delete(13):