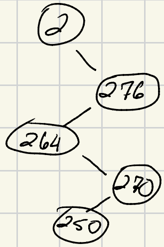
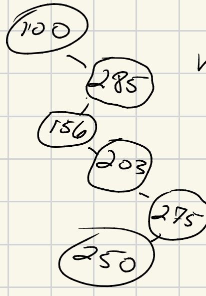
# HW 2 Questions

1. a)



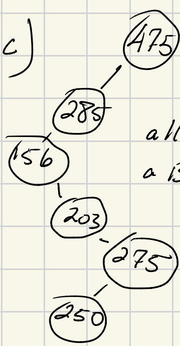

X. This is impossible. $250<264$ meaning that 250 should be in 264's left subtree instead of the right.

b)



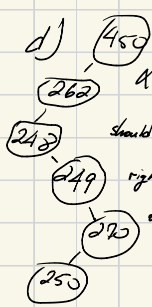

✓. This follows the rules of a BST.

c)




✓. Follows all the rules of a BST

d)




X. 270 should be in 262's right subtree instead of the left subtree.

2. (R-3.11) On average, the running time will be $O(\log n)$.

3. Proper Binary Tree means each parent has 2 children. Therefore, Level 0 = 1 node; level 1 = 2 nodes Level 2 = 4 nodes ∴ at level $i$ there are $2^i$ nodes.

$$1+2+2^2+\dots+2^h=\sum_{i=0}^{h}2^i$$

This is a geometric series ∴

$$\text{Total nodes} = \frac{2^{h+1}-1}{2-1} = 2^{h+1}-1$$

So $2^{h+1}-1$ is the maximum number of nodes

When looking for the minimum, consider the most unbalanced scenario, where the tree will have a single path down to a single leaf. Each height level will have a single internal node that splits into 2 child nodes.

height 0 = 1 node
height 1 = 3 nodes
height 2 = 5 nodes
∴ total nodes $=2h+1$

$$2h+1\le n\le 2^{h+1}-1$$

4. The minimum height is found when the tree is most balanced, meaning a complete binary tree. As proved in question 3, the total number of nodes, $n$, in a complete tree is $n=2^{h+1}-1$.

Solve for $h$:

$$n=2^{h+1}-1$$
$$\log(n+1)=\log_2(2^{h+1})$$
$$\log(n+1)=h+1$$
$$h=\log(n+1)-1$$

To find the maximum, we need an unbalanced tree. As proved in question 3, the total number of nodes, $n$, for an unbalanced tree is $n=2h+1$.

Solve for $h$:

$$n=2h+1$$
$$n-1=2h$$
$$h=\frac{n-1}{2}$$

$$\therefore \log(n+1)-1\le h\le\frac{n-1}{2}$$