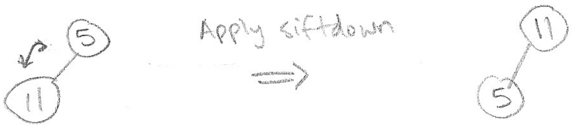
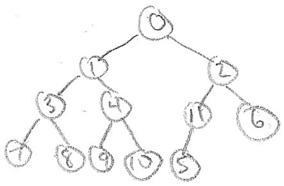
CSCD 320 - HW5

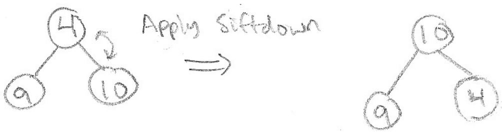
1) We identify the first subtree in a reversed breath-first order



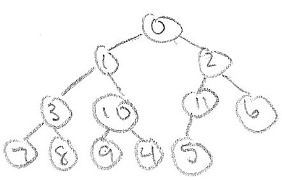
So original tree becomes



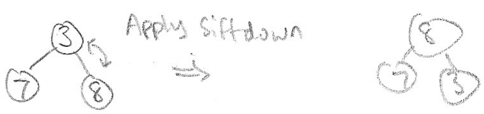
2) Second subtree in a reversed breath-first order



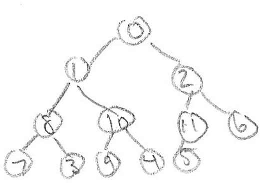
So original tree becomes



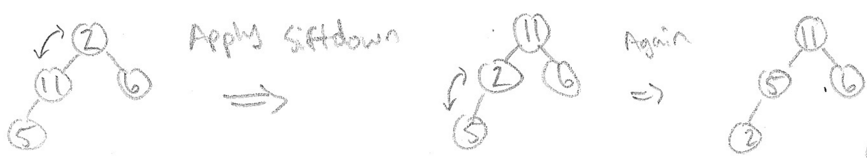
3) Third subtree in a reversed breath first order



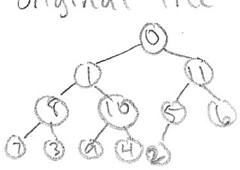
So original tree becomes



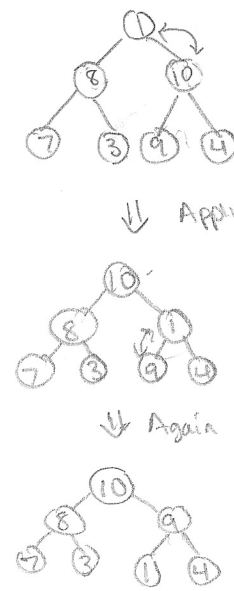
4) Fourth subtree in a reversed breath first order



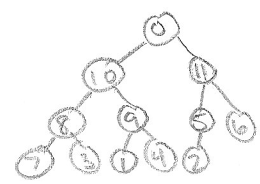
Original Tree becomes



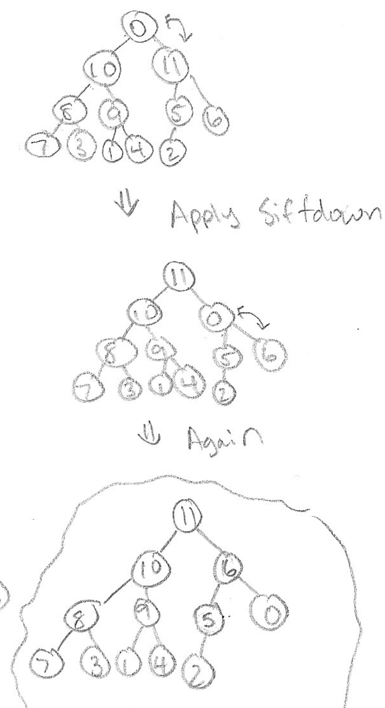
5) Fifth subtree in reversed breath first order



So Original Tree becomes



6) Sixth subtree in reversed breath first order



Done

Theorem: The number of leaves in a non-empty full binary tree is one more than the number of internal nodes.

Base Case: # of nodes in a full binary tree is $\sum_{i=0}^{n} 2^n = 2^0 + 2^1 + 2^2 + \dots + 2^n$, where n is the # of levels in the tree.

Induction Hypothesis: We are to see that the last level of a tree is 1 more than the sum of all upper levels, so we get

$$2^n - \sum_{i=0}^{m} 2^m = 1, \text{ where } m = n-1$$

Induction Step: For a tree with 2 levels (zero indexing, so n is 1)

$$2^n = 2^1 = 2$$

$$\sum_{i=0}^{m} 2^m = 2^0 = 1$$

$$\Rightarrow \quad 2^n - \sum_{i=0}^{m} 2^m = 2^1 - 2^0 = 2 - 1 = 1$$

Therefore, the number of leaves in a non-empty full binary tree is one more than the number of internal nodes (or parent nodes).