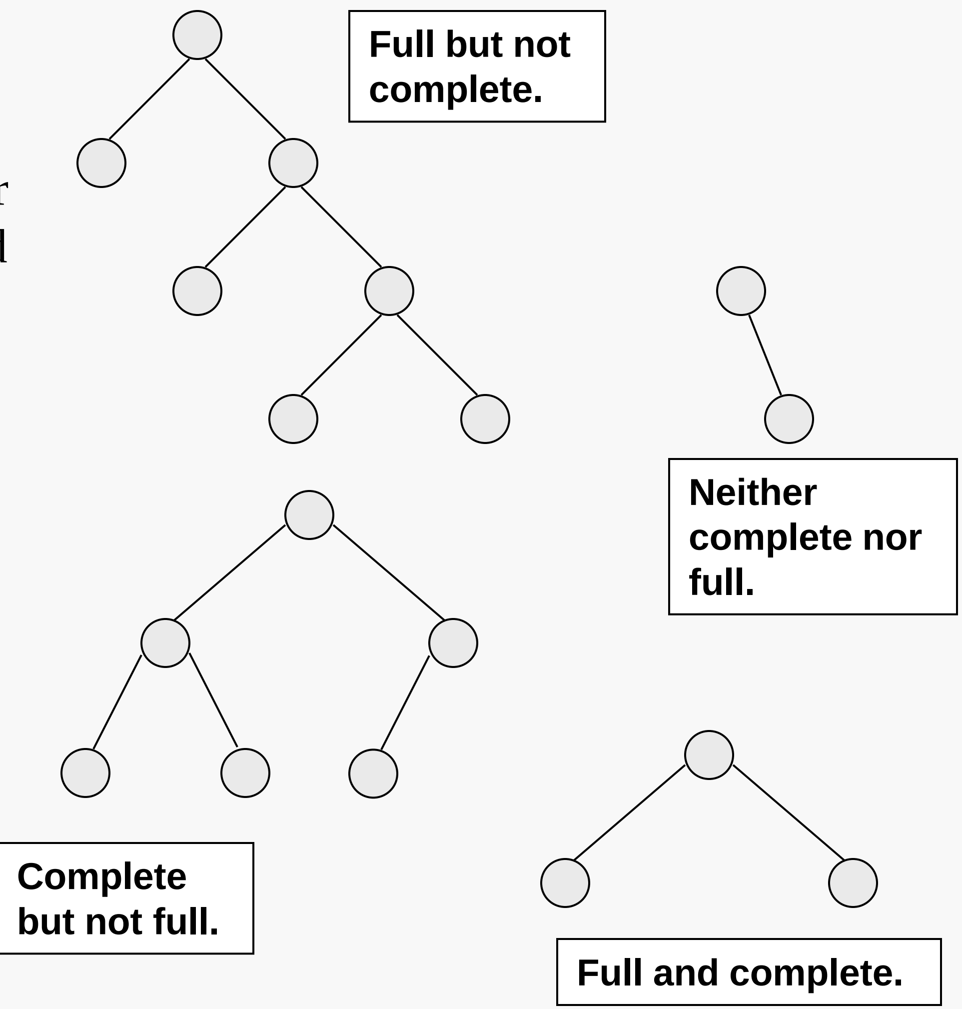
Full and Complete Binary Trees

Here are two important types of binary trees. Note that the definitions, while similar, are logically independent.

Definition: a binary tree T is *full* if each node is either a leaf or possesses exactly two child nodes.

Definition: a binary tree T with n levels is *complete* if all levels except possibly the last are completely full, and the last level has all its nodes to the left side.



Full Binary Tree Theorem

Theorem: Let T be a nonempty, full binary tree Then:

(a) If T has I internal nodes, the number of leaves is L = I + 1.

(b) If T has I internal nodes, the total number of nodes is N = 2I + 1.

(c) If T has a total of N nodes, the number of internal nodes is I = (N – 1)/2.

(d) If T has a total of N nodes, the number of leaves is L = (N + 1)/2.

(e) If T has L leaves, the total number of nodes is N = 2L – 1.

(f) If T has L leaves, the number of internal nodes is I = L – 1.

Basically, this theorem says that the number of nodes N, the number of leaves L, and the number of internal nodes I are related in such a way that if you know any one of them, you can determine the other two.

Proof of Full Binary Tree Theorem

proof of (a): We will use induction on the number of internal nodes, I. Let S be the set of all integers $I \geq 0$ such that if T is a full binary tree with I internal nodes then T has I + 1 leaf nodes.

For the base case, if I = 0 then the tree must consist only of a root node, having no children because the tree is full. Hence there is 1 leaf node, and so $0 \in S$.

Now suppose that for some integer $K \geq 0$, every I from 0 through K is in S. That is, if T is a nonempty full binary tree with I internal nodes, where $0 \leq I \leq K$, then T has I + 1 leaf nodes.

Let T be a full binary tree with K + 1 internal nodes. Then the root of T has two subtrees L and R; suppose L and R have I_L and I_R internal nodes, respectively. Note that neither L nor R can be empty, and that every internal node in L and R must have been an internal node in T, and T had one additional internal node (the root), and so $K + 1 = I_L + I_R + 1$.

Now, by the induction hypothesis, L must have I_L+1 leaves and R must have I_R+1 leaves. Since every leaf in T must also be a leaf in either L or R, T must have $I_L + I_R + 2$ leaves.

Therefore, doing a tiny amount of algebra, T must have K + 2 leaf nodes and so $K + 1 \in S$. Hence by Mathematical Induction, $S = [0, \infty)$.

QED

Limit on the Number of Leaves

Theorem: Let T be a binary tree with λ levels. Then the number of leaves is at most $2^{\lambda-1}$.

proof: We will use strong induction on the number of levels, λ. Let S be the set of all integers $\lambda \geq 1$ such that if T is a binary tree with λ levels then T has at most $2^{\lambda-1}$ leaf nodes.

For the base case, if $\lambda = 1$ then the tree must have one node (the root) and it must have no child nodes. Hence there is 1 leaf node (which is $2^{\lambda-1}$ if $\lambda = 1$), and so $1 \in S$.

Now suppose that for some integer $K \geq 1$, all the integers 1 through K are in S. That is, whenever a binary tree has M levels with $M \leq K$, it has at most 2^{M-1} leaf nodes.

Let T be a binary tree with K + 1 levels. If T has the maximum number of leaves, T consists of a root node and two nonempty subtrees, say S_1 and S_2. Let S_1 and S_2 have M_1 and M_2 levels, respectively. Since M_1 and M_2 are between 1 and K, each is in S by the inductive assumption. Hence, the number of leaf nodes in S_1 and S_2 are at most 2^{K-1} and 2^{K-1}, respectively. Since all the leaves of T must be leaves of S_1 or of S_2, the number of leaves in T is at most $2^{K-1} + 2^{K-1}$ which is 2^K. Therefore, K + 1 is in S.

Hence by Mathematical Induction, $S = [1, \infty)$.

QED

More Useful Facts

Theorem: Let T be a binary tree. For every $k \geq 0$, there are no more than 2^k nodes in level k.

Theorem: Let T be a binary tree with λ levels. Then T has no more than $2^\lambda - 1$ nodes.

Theorem: Let T be a binary tree with N nodes. Then the number of levels is at least $\lceil \log (N + 1) \rceil$.

Theorem: Let T be a nonempty binary tree with L leaves. Then the number of levels is at least $\lceil \log L \rceil + 1$.

A Note on Terminology

The definition of the *height* of a tree varies by author.

Traditional usage is that the height is the number of levels in the tree.

Accordingly, a tree with one node has height 1 and an empty tree has height 0.

Some authors now say that the height is the number of edges in the longest path from the root of the tree to a leaf.

Accordingly, a tree with one node has height 0 and an empty tree has height ??.

FWIW, I consider the second usage to be absurd.

In this course, we will refer to the number of levels (of nodes) in the tree rather than the height of the tree.