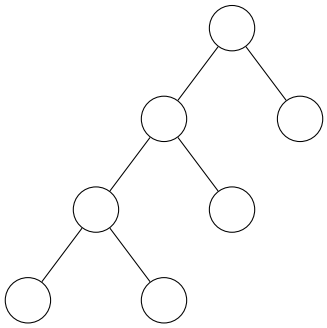
Leaves vs. Internal Nodes in a Full Binary Tree

Definition 1. A **binary tree** is called **full** if every node has either 0 or 2 children.

Example. The following is a full binary tree with 3 internal nodes and 4 leaves:



Theorem 1. *In a non-empty full binary tree (FBT), the number of leaves is always exactly* 1 *more than the number of internal nodes.*

Proof. We proceed by strong induction on n, the number of internal nodes.

Let $L(n)$ denote the number of leaves in a non-empty FBT with exactly n internal nodes. We wish to show that $L(n) = n + 1$ for all $n \geq 0$.

Base case $(n = 0)$.
A tree consisting of a single node has no internal nodes and exactly 1 leaf. Therefore

$$L(0) = 1 = 0 + 1.$$

Inductive hypothesis (I.H.).
Assume that $L(i) = i + 1$ holds for every i with $0 \leq i < n$.

Inductive step.
Let T be a non-empty FBT with $n \geq 1$ internal nodes. Because T is full and non-trivial, it contains at least one internal node, which must have exactly two children; hence T has at least two leaves. In particular, there exist *sibling* leaves — two leaves that share the same parent.

1. **Remove the two sibling leaves.** Let T' be the FBT obtained from T by deleting these two sibling leaves. Their former parent, which was an internal node in T, becomes a leaf in T'. Thus T' has $n - 1$ internal nodes.

2. **Apply the inductive hypothesis.** Since $n - 1 < n$, the I.H. applies to T':

$$L(n-1) = (n-1) + 1 = n.$$

   Hence T' has exactly n leaves.

3. **Re-attach the two sibling leaves.** Restoring the two removed leaves converts their parent from a leaf back into an internal node (net change: -1 leaf) and introduces two new leaves (net change: $+2$ leaves). Therefore

$$L(n) = n - 1 + 2 = n + 1.$$

This completes the induction. We conclude that in every non-empty FBT, the number of leaves equals the number of internal nodes plus one. □