# **Binary Tree Height**

* Trees come in many shapes.
* Each binary tree in Figure 15-5 contains the same number of nodes, though their structures are quite different.

Chart, line chart

Description automatically generated

* Although each of these trees has seven nodes, some are “taller” than others.

## **Height Formal Definition**

* The **height** of a tree is the **number of nodes** on the **longest path** from the **root to a leaf**.

## **Level of Node n**

* There are other equivalent ways to define the height of a tree T.
* One way uses the following definition of the level of a node n:
  + If n is the root of T, it is at level 1.
  + If n is not the root of T, its level is 1 greater than the level of its parent.
* For example, in Figure 15-5a
  + node A is at level 1,
  + node B is at level 2,
  + node D is at level 3.
* The height of a tree T in terms of the levels of its nodes is defined as follows:
  + If T is empty, its height is 0.
  + If T is not empty, its height is equal to the maximum level of its nodes.
* By applying this definition to the trees in Figure 15-5, you will find that their heights are, respectively, 3, 5, and 7, as was stated earlier.

**Recursive Definition of Height**

* For binary trees, it is often convenient to use an equivalent recursive definition of height:
* **T is a binary tree if either**
  + If *T* is empty, its height is 0.
  + If *T* is a nonempty binary tree, then because *T* is of the form *r*

*r*

page455image1660818112page455image1660819840

*TL*  *TR*

the height of *T* is 1 greater (the +1 is the root) than the max height of its root’s taller subtree:

***height* (*T*) = 1 + *max*{*height* (*T*L), *height* (*T*R)}**

* Remember,
  + The depth of a tree is equal to the depth of the deepest leaf
  + The height of a tree is equal to the height of the root.
* Therefore, the depth and height of the tree are always equal.
  + the height of the tree is equal to the depth of the tree
  + the maximum depth is equal to the maximum height.

# **Full Binary Tree**

* In a **full binary tree** of height *h*, all nodes that are at a **level less than *h***have two children each.
* Each node in a full binary tree has no nodes with one child; they all have zero or two.
* **T is a full binary tree if either**
  + All nodes at a level less than *h* have two children each
  + All leaves (zero children) are filled at height *h*
* Each node in a full binary tree has left and right subtrees of the same height.
* Among binary trees of height *h*, a full binary tree has as many leaves as possible, and they all are at level *h*.
* Intuitively, **a full binary tree has no missing nodes**.
* Figure 15-6 depicts a full binary tree of height 3.

Chart, line chart

Description automatically generated

* When proving properties about full binary trees—such as how many nodes they have—the following recursive definition of a full binary tree is convenient:
  + If *T* is empty, *T* is a full binary tree of height 0.
  + If *T* is not empty and has height *h* > 0, *T* is a full binary tree if its root’s subtrees are both full
  + If *T* is a nonempty binary tree and has height *h* > 0, then because *T* is of the form *r*

*r*

page455image1660818112page455image1660819840

*TL*  *TR*

*T* is a full binary tree if its root’s subtrees are both full binary trees of height *h* – 1.