Introduction to Trees

Section 11.1

Section Summary

- Introduction to Trees
- Rooted Trees
- Trees as Models
- Properties of Trees

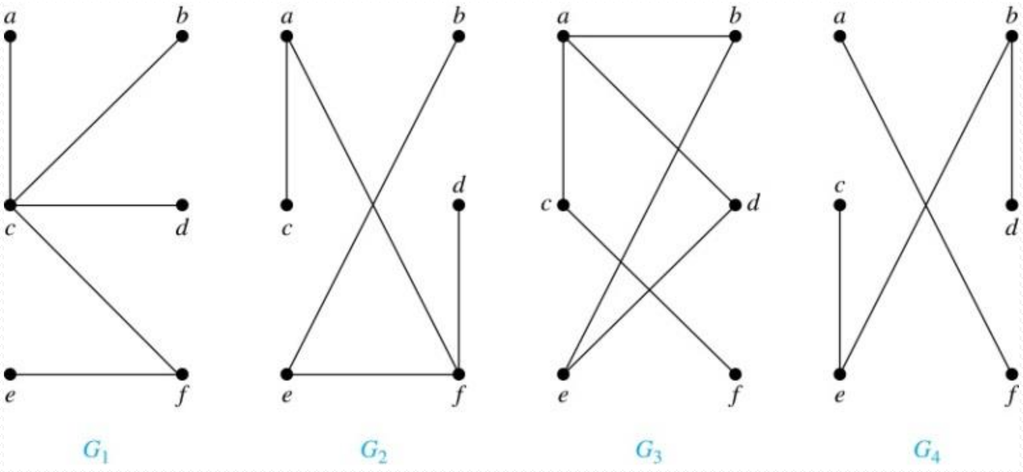
Trees

Definition: A *tree* is a connected undirected graph with no simple circuits.

Example: Which of these graphs are trees?

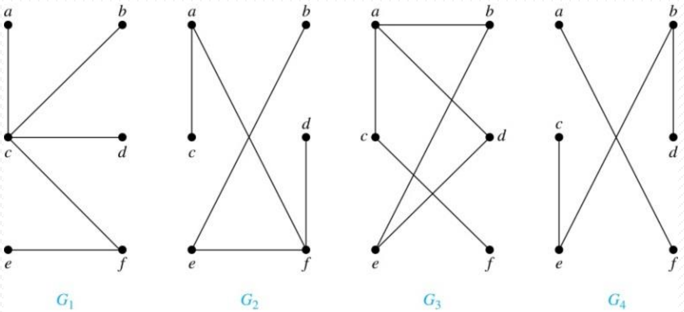
CA:

a) G_1 and G_2
b) G_2 and G_3
c) G_1 and G_4



Trees

Definition: A *tree* is a connected undirected graph with no simple circuits.



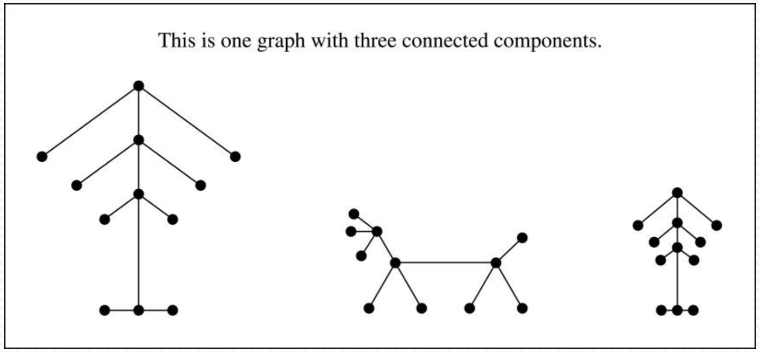
Example: Which of these graphs are trees?

Solution: G_1 and G_2 are trees - both are connected and have no simple circuits. Because e, b, a, d, e is a simple circuit, G_3 is not a tree. G_4 is not a tree because it is not connected.

Trees

Definition: A *tree* is a connected undirected graph with no simple circuits.

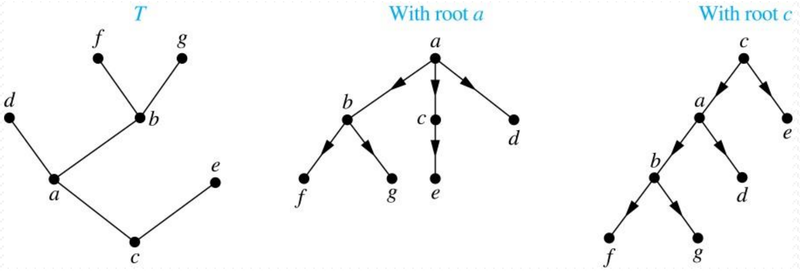
Definition: A *forest* is a graph that has no simple circuit, but is not connected. Each of the connected components in a forest is a tree.



Rooted Trees

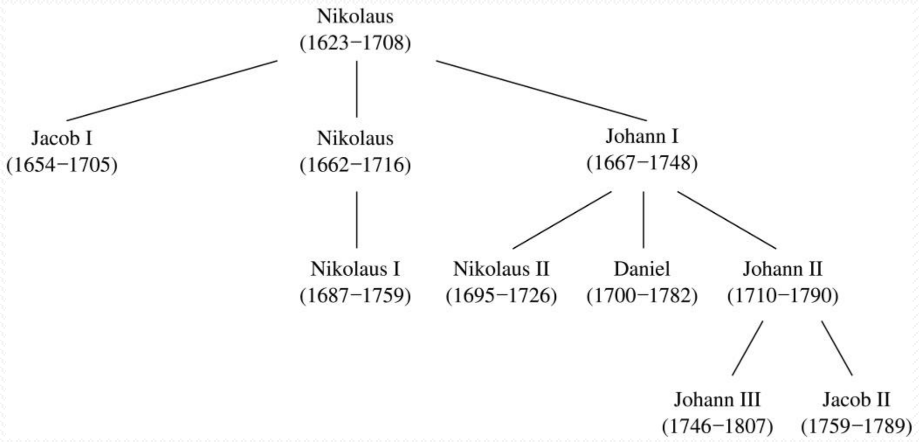
Definition: A *rooted tree* is a tree in which one vertex has been designated as the *root* and every edge is directed away from the root.

An unrooted tree is converted into different rooted trees when different vertices are chosen as the root.



Rooted Tree Terminology

- Terminology for rooted trees is a mix from botany and genealogy (such as this family tree of the Bernoulli family of mathematicians).



- If v is a vertex of a rooted tree other than the root, the *parent* of v is the unique vertex u such that there is a directed edge from u to v. When u is a parent of v, v is called a *child* of u. Vertices with the same parent are called *siblings*.
- The *ancestors* of a vertex are the vertices in the path from the root to this vertex, excluding the vertex itself and including the root. The *descendants* of a vertex v are those vertices that have v as an ancestor.
- A vertex of a rooted tree with no children is called a *leaf*. Vertices that have children are called *internal vertices*.
- If a is a vertex in a tree, the *subtree* with a as its root is the subgraph of the tree consisting of a and its descendants and all edges incident to these descendants.

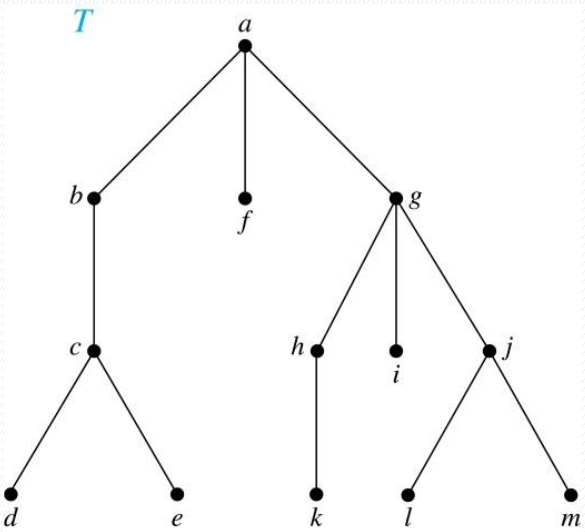
Terminology for Rooted Trees

Example: In the rooted tree T (with root a):

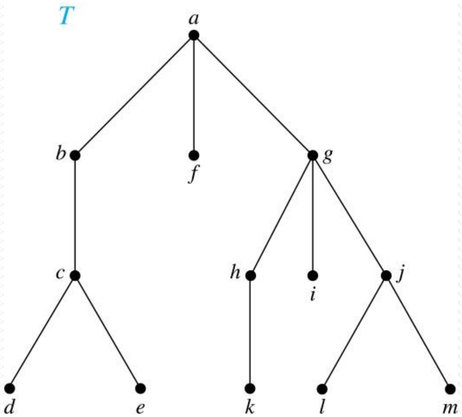
(i) Find the parent of c, the children of g, the siblings of h, the ancestors of e, and the descendants of b.

(ii) Find all internal vertices and all leaves.

(iii) What is the subtree rooted at G?



Terminology for Rooted Trees

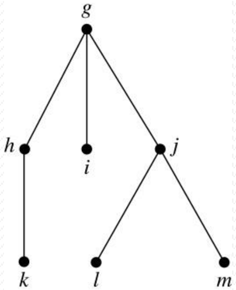


Solution:

(i) The parent of c is b. The children of g are h, i, and j. The siblings of h are i and j. The ancestors of e are c, b, and a. The descendants of b are c, d, and e.

(ii) The internal vertices are a, b, c, g, h, and j. The leaves are d, e, f, i, k, l, and m.

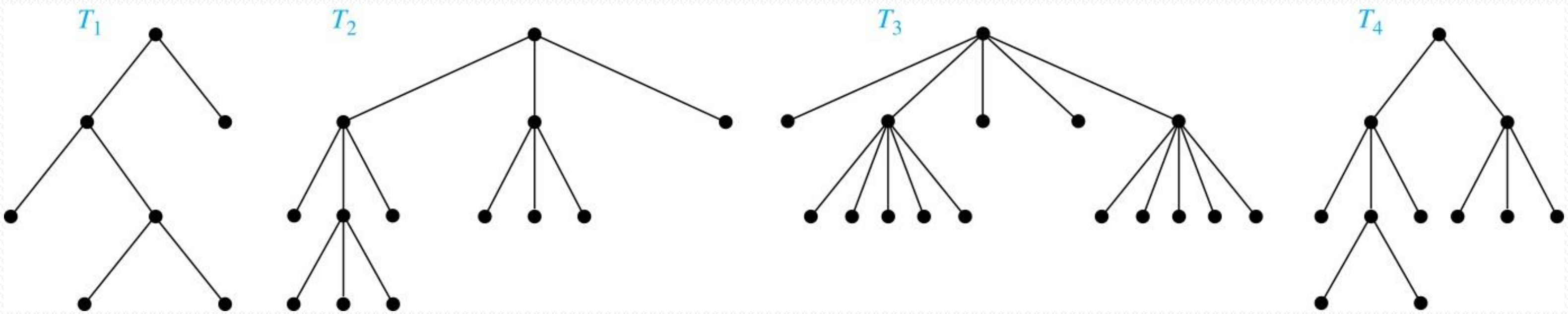
(iii) We display the subtree rooted at g.



m-ary Rooted Trees

Definition: A rooted tree is called an *m-ary tree* if every internal vertex has no more than m children. The tree is called a *full m-ary tree* if every internal vertex has exactly m children. An m-ary tree with m = 2 is called a *binary* tree.

Example: Which of the rooted trees are NOT full m-ary trees?



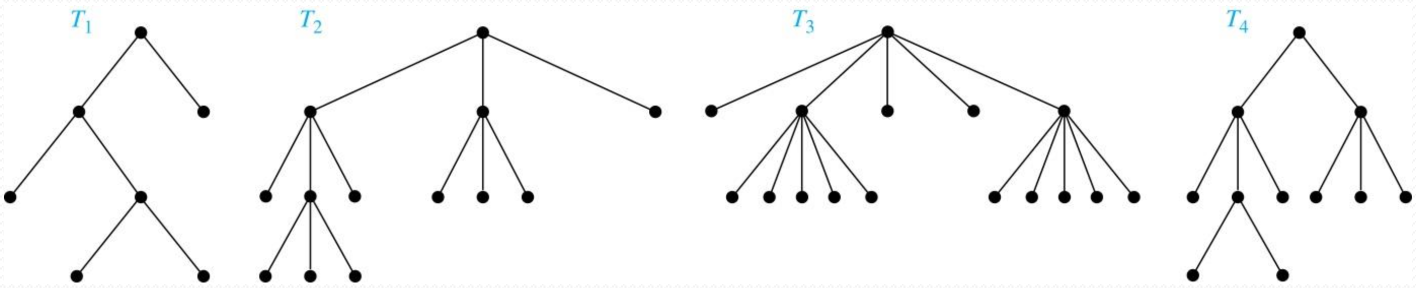
Cl.A:

a) T_1
b) T_2
c) T_3
d) T_4

m-ary Rooted Trees

Definition: A rooted tree is called an *m-ary tree* if every internal vertex has no more than m children. The tree is called a *full m-ary tree* if every internal vertex has exactly m children. An m-ary tree with $m = 2$ is called a *binary* tree.

Example: Which of the rooted trees are NOT full m-ary trees?



Solution: T_1 is a full binary tree because each of its internal vertices has two children. T_2 is a full 3-ary tree because each of its internal vertices has three children. In T_3 each internal vertex has five children, so T_3 is a full 5-ary tree. **T_4 is not a full m-ary tree for any m because some of its internal vertices have two children and others have three children.**

Ordered Rooted Trees

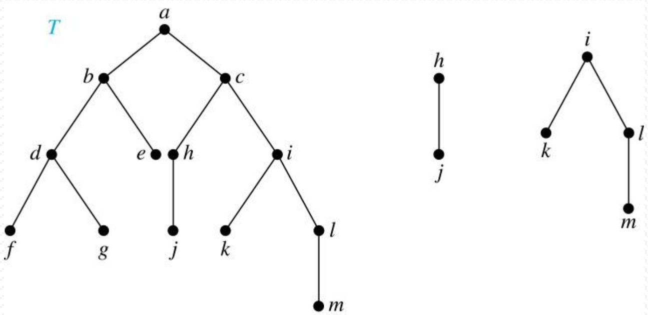
Definition: An *ordered rooted tree* is a rooted tree where the children of each internal vertex are ordered (from left to right).

Definition: A *binary tree* is an ordered rooted where each internal vertex has at most two children. If an internal vertex of a binary tree has two children, the first is called the *left child* and the second the *right child*.
The tree rooted at the left child of a vertex is called the *left subtree* of this vertex, and the tree rooted at the right child of a vertex is called the *right subtree* of this vertex.

Example: Consider the binary tree T.

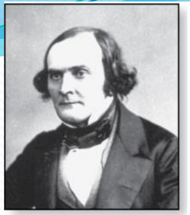
(*i*) What are the left and right children of d?

(*ii*) What are the left and right subtrees of c?



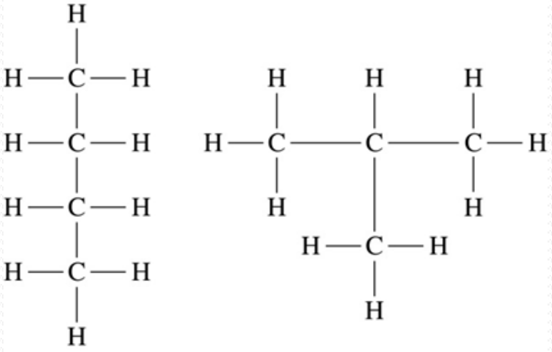
(a) (b) (c)

Arthur Cayley
(1821-1895)

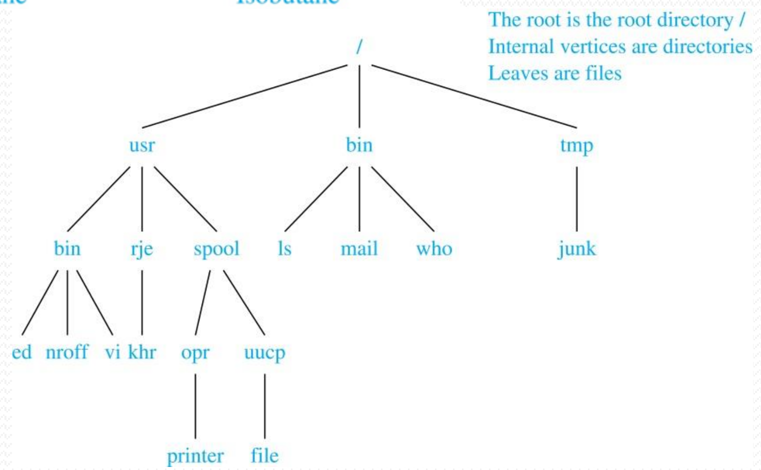


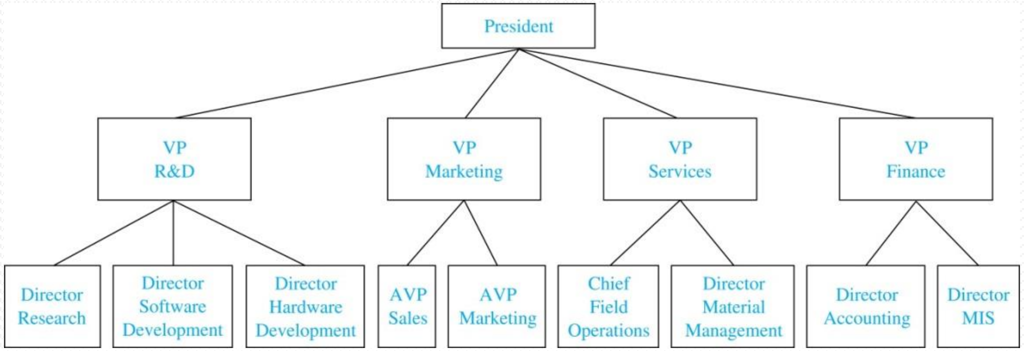
Trees as Models

- Trees are used as models in computer science, chemistry, geology, botany, psychology, and many other areas.
- Trees were introduced by the mathematician Cayley in 1857 in his work counting the number of isomers of saturated hydrocarbons. The two isomers of butane are shown at the right.
- The organization of a computer file system into directories, subdirectories, and files is naturally represented as a tree.
- Trees are used to represent the structure of organizations.



Butane Isobutane





Properties of Trees

Theorem 2: A tree with n vertices has $n - 1$ edges.

***Proof* (*by mathematical induction*):**

BASIS STEP: When $n = 1$, a tree with one vertex has no edges. Hence, the theorem holds when $n = 1$.

INDUCTIVE STEP: Assume that every tree with k vertices has $k - 1$ edges.

Suppose that a tree T has $k + 1$ vertices and that v is a leaf of T. Let w be the parent of v. Removing the vertex v and the edge connecting w to v produces a tree T' with k vertices. By the inductive hypothesis, T' has $k - 1$ edges. Because T has one more edge than T', we see that T has k edges. This completes the inductive step. ◀

Counting Vertices in Full *m*-Ary Trees

Theorem 3: A full m-ary tree with i internal vertices has $n = mi + 1$ vertices.

Proof : Every vertex, except the root, is the child of an internal vertex. Because each of the i internal vertices has m children, there are mi vertices in the tree other than the root. Hence, the tree contains $n = mi + 1$ vertices. ◀

Counting Vertices in Full *m*-Ary Trees (*continued*)

Theorem 4: A full m-ary tree with

(i) n vertices has $i = (n - 1)/m$ internal vertices and $l = [(m - 1)n + 1]/m$ leaves,

(ii) i internal vertices has $n = mi + 1$ vertices and $l = (m - 1)i + 1$ leaves,

(iii) l leaves has $n = (ml - 1)/(m - 1)$ vertices and $i = (l - 1)/(m - 1)$ internal vertices.

proofs of parts (ii) and (iii) are left as exercises

Proof (***of part i***)**:** Solving for i in $n = mi + 1$ (from Theorem 3) gives $i = (n - 1)/m$. Since each vertex is either a leaf or an internal vertex, $n = l + i$. By solving for l and using the formula for i, we see that

$$l = n - i = n - (n - 1)/m = [(m - 1)n + 1]/m. \quad ◀$$

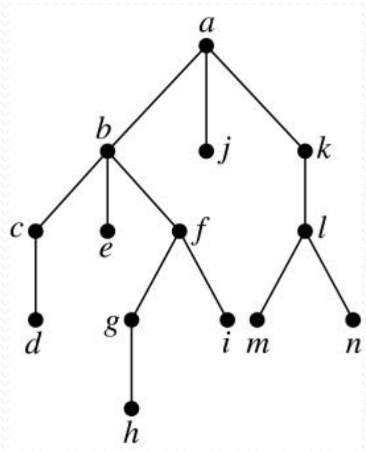
Level of vertices and height of trees

- When working with trees, we often want to have rooted trees where the subtrees at each vertex contain paths of approximately the same length.
- The ***level*** of a vertex *v* in a rooted tree is the length of the unique path from the root to this vertex.
- The ***height*** of a rooted tree is the maximum of the levels of the vertices.

Example:

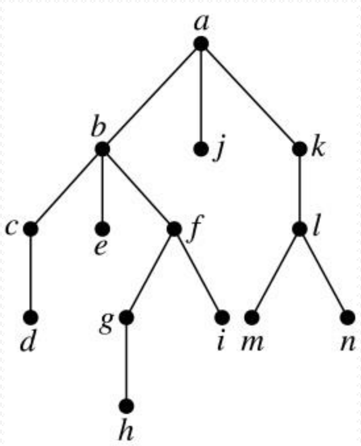
(*i*) Find the level of each vertex in the tree to the right.

(*ii*) What is the height of the tree?



Level of vertices and height of trees

- The ***level*** of a vertex *v* in a rooted tree is the length of the unique path from the root to this vertex.
- The ***height*** of a rooted tree is the maximum of the levels of the vertices.



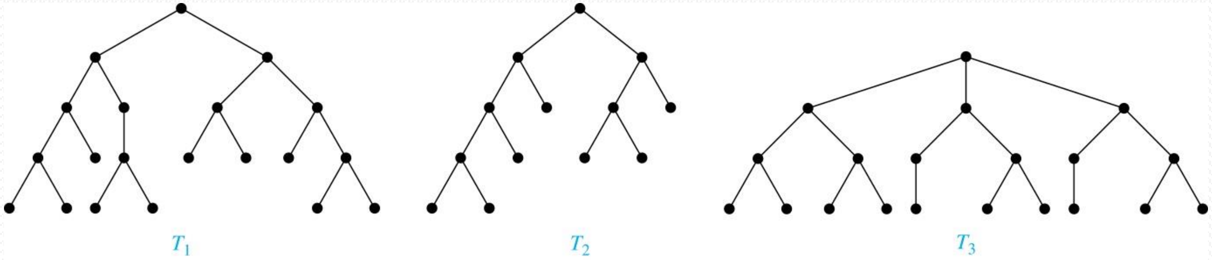
Solution:

- The root *a* is at level 0.
- Vertices *b*, *j*, and *k* are at level 1.
- Vertices *c*, *e*, *f*, and *l* are at level 2.
- Vertices *d*, *g*, *i*, *m*, and *n* are at level 3.
- Vertex *h* is at level 4.
- The height is 4, since 4 is the largest level of any vertex.

Balanced *m*-Ary Trees

Definition: A rooted m-ary tree of height h is *balanced* if all leaves are at levels h or $h - 1$.

Example: Which of the rooted trees shown below is NOT balanced?



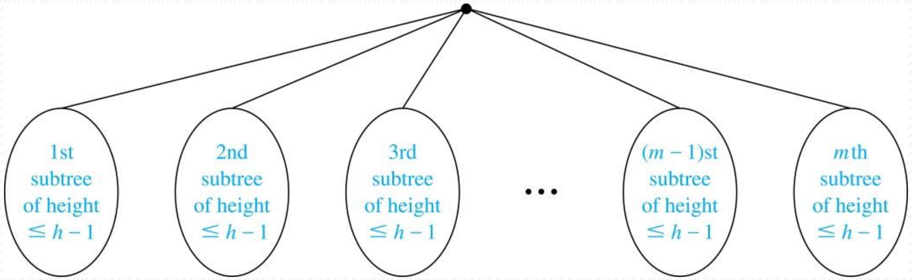
The Bound for the Number of Leaves in an m-ary Tree

Theorem 5: There are at most m^h leaves in an m-ary tree of height h.

***Proof* (*by mathematical induction on height*):**

BASIS STEP: Consider an m-ary trees of height 1. The tree consists of a root and no more than m children, all leaves. Hence, there are no more than $m^1 = m$ leaves in an m-ary tree of height 1.

INDUCTIVE STEP: Assume the result is true for all m-ary trees of height $< h$. Let T be an m-ary tree of height h. The leaves of T are the leaves of the subtrees of T we get when we delete the edges from the root to each of the vertices of level 1.



Each of these subtrees has height $\leq h-1$. By the inductive hypothesis, each of these subtrees has at most m^{h-1} leaves. Since there are at most m such subtees, there are at most $m \cdot m^{h-1} = m^h$ leaves in the tree. ◀

Corollary 1: If an m-ary tree of height h has l leaves, then $h \geq \lceil \log_m l \rceil$. If the m-ary tree is full and balanced, then $h = \lceil \log_m l \rceil$. (*see text for the proof*)