Topic 6: Graphs and Trees

6.1 Introduction

6.2 Terminology

6.3 Representation and Isomorphism

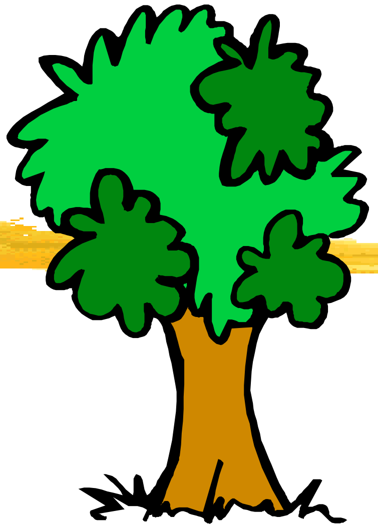
6.4 Connectivity

6.5 Euler Paths

6.6 Hamilton Paths

6.7 Trees

6.7 Trees



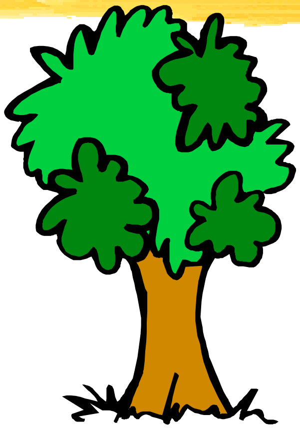
- Introduction to Trees
- Rooted Trees
- Examples
- References:
  - *Rosen* (8th Edition) Section 11.1 – Introduction to Trees (page nos 781-793) ;
  - *Grimaldi* (5th Edition) Section 12.1 - Trees – Definitions, Properties, Examples, (page nos 581-587) & part of 12.2 – Rooted Trees – (page nos 587-605).

Introduction to Trees

- Trees were introduced by the English mathematician Arthur Cayley in 1857 to count certain types of chemical compounds
- Trees are particularly useful in computer science, where they are employed in a wide range of algorithms, such as
  - searching (Depth-First Search, Breadth-First Search)
  - minimum spanning trees (Prim's, Kruskal's algorithm)
  - game trees (Chess, Tic-Tac-Toe)
  - coding (Huffman Coding)

Introduction to Trees

What is a Tree?



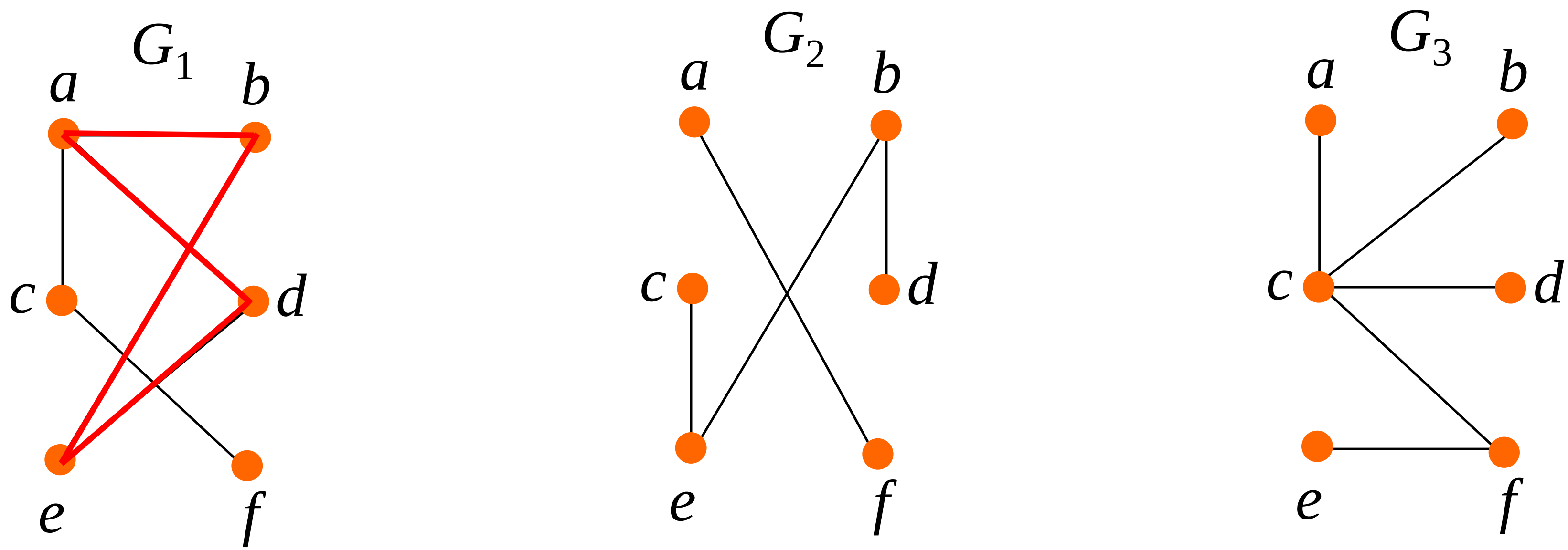
Definition:

A **tree** is a connected undirected graph with no (simple) circuits.

Remarks: Since a tree cannot have a (simple) circuit, a tree cannot contain multiple edges or loops. Therefore, any tree must be a simple graph.

Introduction to Trees

Example: Which of the following graphs is tree?



No! (simple circuit a,b,e,d,a)

No! (not connected)
(no path from vertex a to vertex b; therefore, not connected)

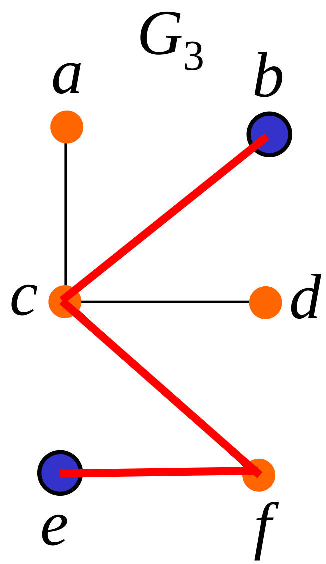
Yes! (connected and no simple circuits)

Introduction to Trees

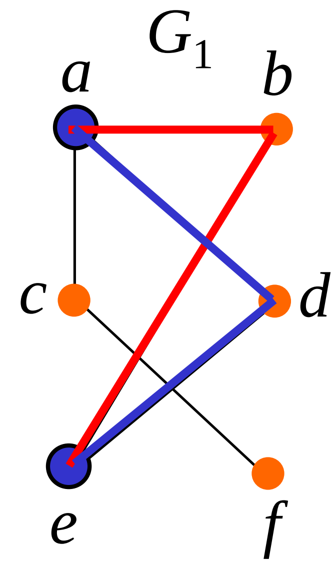
<u>Theorem 1:</u> (Alternative definition of a tree)

An undirected graph is a tree if and only if there is a unique simple path between any two vertices.

<u>Example:</u>



Exactly one path between any two vertices. Thus, G_3 is a Tree



Two paths between a and e. Thus, G_2 is not Tree. (These two paths can be used to form simple circuit.)

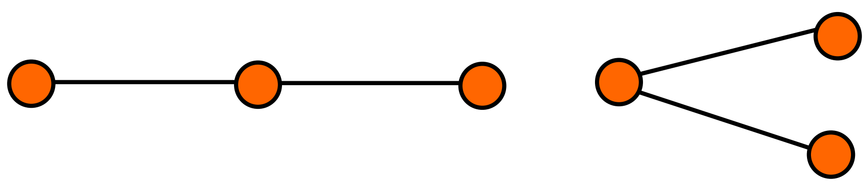
Introduction to Trees

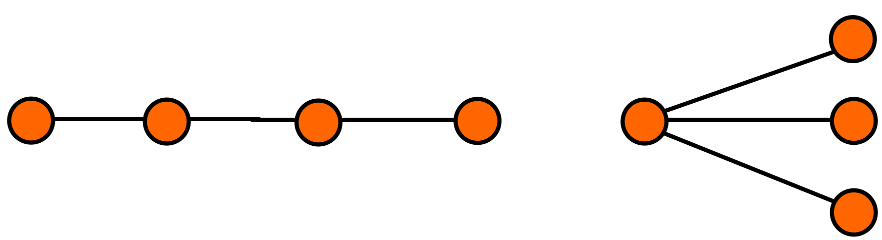
<u>Question:</u> How many edges has a tree with *n* vertices?

<u>Examples first:</u>

1 vertex:  $\Rightarrow$ 0 edges

2 vertices:  $\Rightarrow$ 1 edge

3 vertices:  $\Rightarrow$ 2 edges

4 vertices:  $\Rightarrow$ 3 edges

Lemma 1: Let *T* be a tree (i.e., connected graph). *T* has at least one vertex *v* having degree 1.

Proof: (sketch) Suppose *T* has all vertices of degree 2 or more and it is connected graph. Then, no matter how we draw this tree, it would always have a cycle, which contradicts the definition of tree.

Lemma 2: Let *T* be a tree (i.e., connected graph). *T* has at least two vertices, say *v*, *w* having degree 1.

Introduction to Trees

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

Proof: (sketch) by mathematical induction.

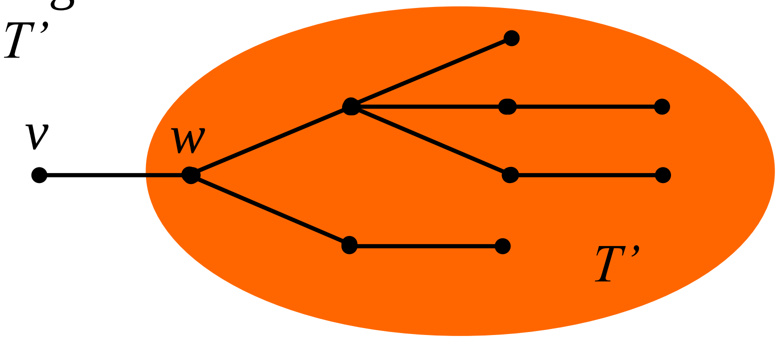
Basic Step (n=1): A tree with one vertex has zero edges.

Inductive Hypothesis (n=k): Assume that any tree with k vertices has k–1 edges.

Inductive Step (n=k+1): Let T be a tree with k+1 vertices (to show: T has k edges).

Lemma 1 ensures that T has at least one vertex v having degree 1. Let v be such vertex in T and w is other node where {w, v} is edge.

Removing v and {v,w} from T, produces a tree T' with k vertices.



According to the Inductive Hypothesis T' has k–1 edges.

Thus, T has k edges since it T has exactly one edge more than T'.

Tree: Some characterizations

Theorem 1.8. *Let T be a graph with n vertices. Then the following statements are equivalent.*

(1) *T is a tree.*
(2) *T contains no cycles and has $n-1$ edges.*
(3) *T is connected and has $n-1$ edges.*
(4) *T is connected, and every edge is a cut-edge.*
(5) *Any two vertices of T are connected by exactly one path.*
(6) *T contains no cycles, and for any new edge e, the graph $T+e$ has exactly one cycle.*

Rooted Trees

Definition:

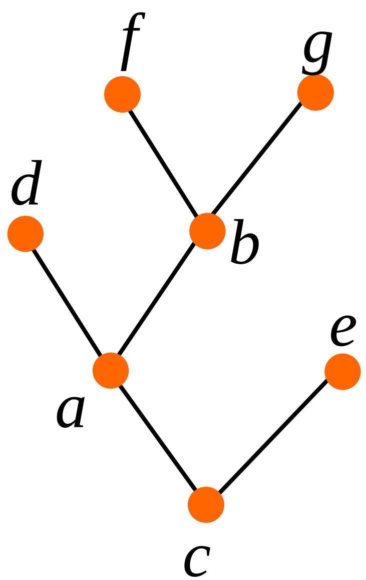
A **rooted tree** is a tree in which one vertex has been designated as the root and every edge is directed away from the root.

Remarks:

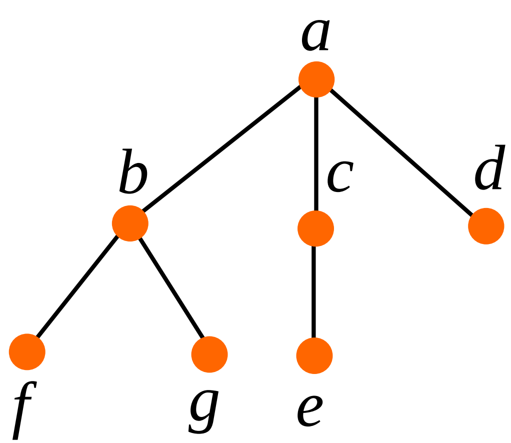
- We can change an unrooted tree to a rooted tree by choosing any vertex as the root.
- Different choices of the root produce different rooted trees
- We usually draw a rooted tree with its root at the top of the graph
- Arrows indicating the directions of the edges in a rooted tree can be omitted (since the choice of the root determines the direction of the edges)

Rooted Trees

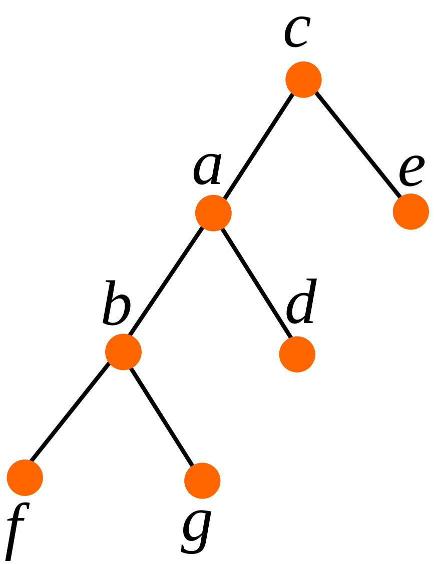
Example:



Unrooted Tree T



Rooted Tree formed from T taking a as the root



Rooted Tree formed from T taking c as the root

Rooted Trees

Terminology: Let T be a rooted tree.

• If v is a vertex of T 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 the parent of v, v is a **child** of u

• Vertices with the same parent are called **siblings**

• The **ancestors** of a vertex other than the root are the vertices in the path from the root to this vertex, excluding the vertex itself, including the root.

• The **descendants** of v are those vertices that have v as an ancestor.

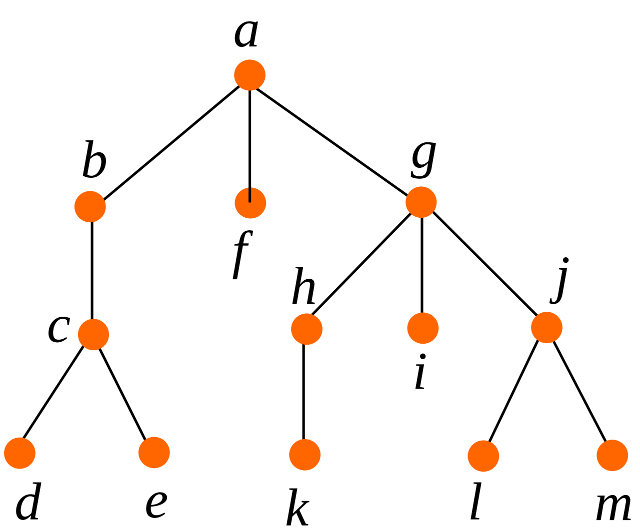
• A vertex is called a **leaf**, if it has no children

• Vertices that have children are called **internal vertices**

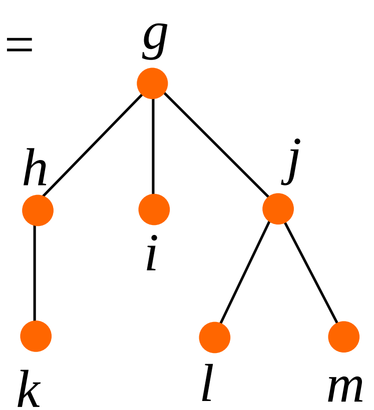
• If a is a vertex, the **subtree** with a as its root is the subgraph of T consisting of a and its descendants and all edges incident to these descendants.

Rooted Trees

<u>Example:</u> Given the following rooted Tree:



- parent of $c = b$
- children of $g = h, i, j$
- siblings of $h = i, j$
- ancestors of $e = c, b, a$
- descendants of $b = c, d, e$
- internal vertices $= a, b, c, g, h, j$
- leaves $= d, e, f, k, i, l, m$
- subtree rooted at $g =$



Rooted Trees

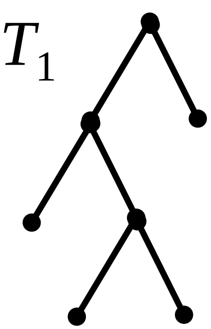
<u>Definition:</u> A rooted tree is called ***<u>m</u>*** **<u>-ary tree</u>** if every internal vertex has no more than *m* children.

The tree is called a **<u>full *m*-ary tree</u>** if every internal vertex has exactly m children.

An *m*-ary tree with *m*=2 is called a **<u>binary tree</u>**.

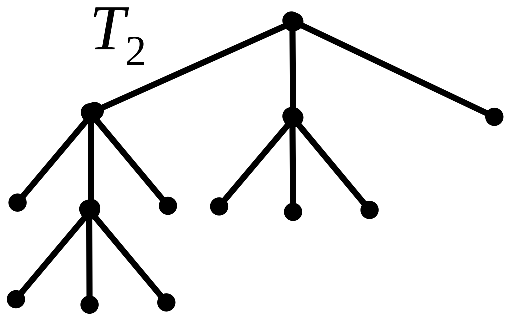
<u>Example:</u> Which of the following rooted trees are full *m*-ary trees?

T_1



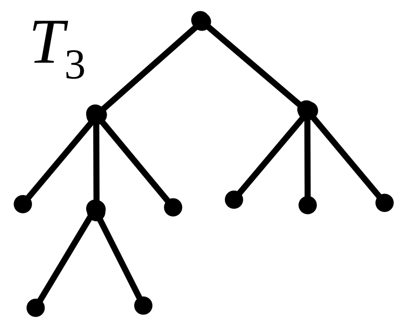
T_1 is a full binary tree

T_2



T_2 is a full 3-ary tree

T_3

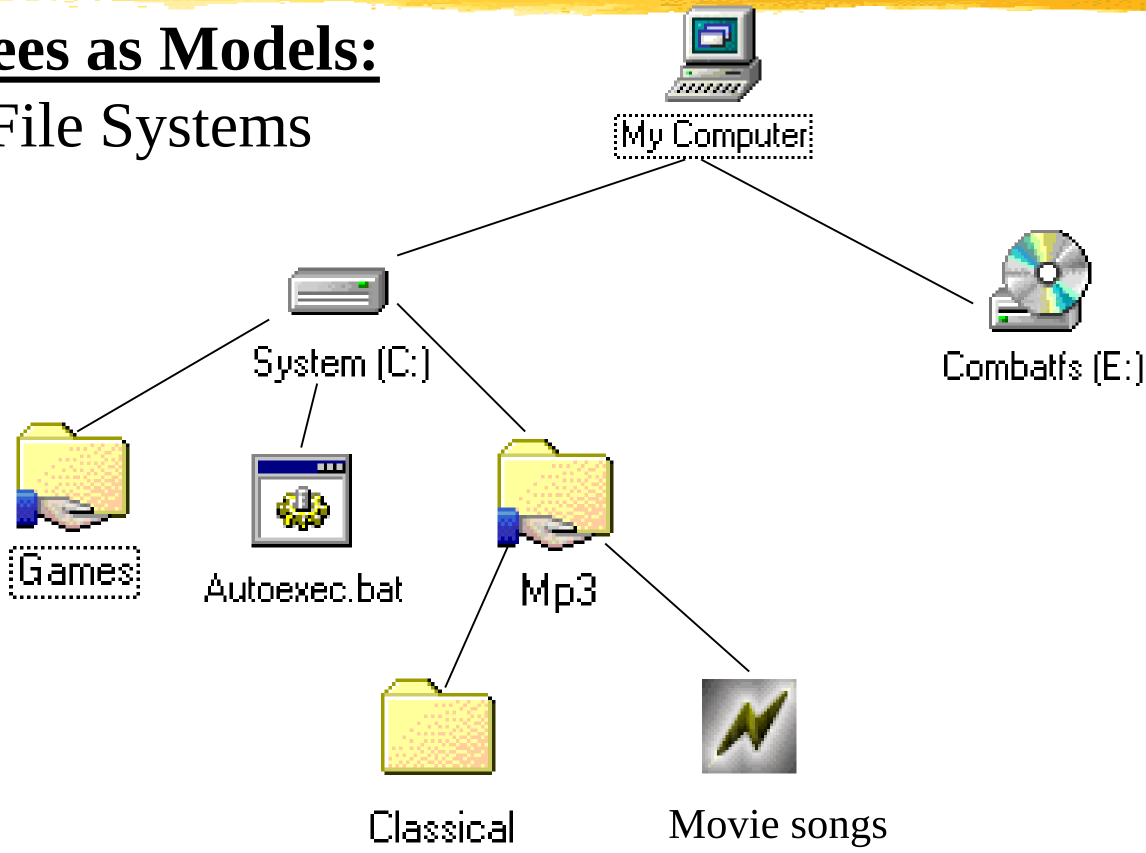


T_3 is not a full *m*-ary tree

Rooted Trees

Rooted Trees as Models:

Computer File Systems



Root represents root directory; internal vertices represent subdirectories; leaves represent ordinary files or empty directories

Rooted Trees

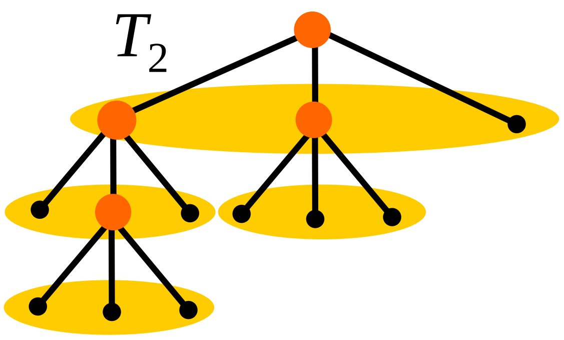
Question: How many vertices has a full m-ary tree with i internal vertices?

Example: T_2 is full (m=3)-ary tree with i=4 internal vertices.

Each internal vertex has m=3 children.

Each vertex except the root is the child of an internal vertex.

Thus, there $m \times i$ plus the root vertices, i.e. $m \times i + 1$.



Theorem 3:

A full m-ary tree with i internal vertices contains $n = m \cdot i + 1$ vertices.

Rooted Trees

Is there a full binary tree that has 10 internal vertices and 13 leaves?

No!

By Theorem 3 a full binary tree with 10 internal vertices has $2\times10 + 1 = 21$ vertices and therefore 11 leaves, not 13!

Rooted Trees

Definition:

The **level** of a vertex v in a rooted tree is the length of the unique path from the root to v.

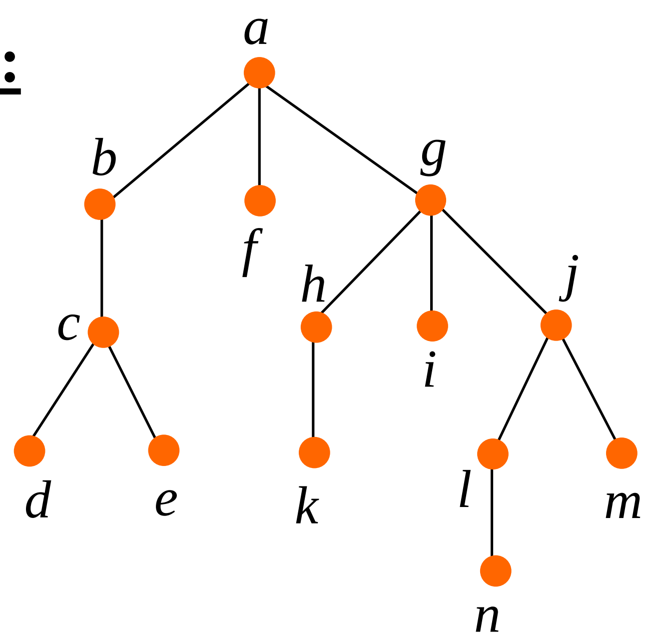
The **level of the root** is defined to be zero.

The **height** of a rooted tree is the maximum level of its vertices.

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

Rooted Trees

Example:

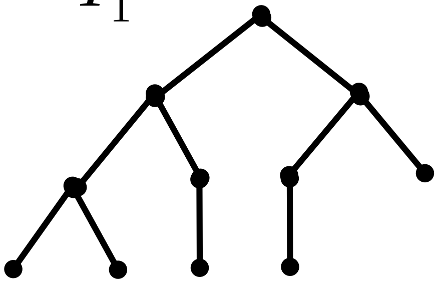


- a is at level 0
- b, f, g are at level 1
- c, h, i, j are at level 2
- d, e, k, l, m are at level 3
- n is at level 4
- the height of this tree is 4

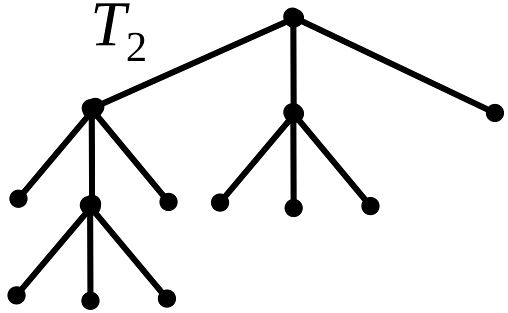
Rooted Trees

Example: Which of the following rooted trees are balanced?

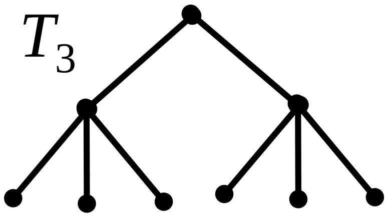
T_1



T_2



T_3



T_1 is balanced (height(T_1)=3 and all leaves at level 2 or 3)

T_2 is not balanced (height(T_1)=3, but leaves at level 1, 2 and 3)

T_3 is balanced (height(T_3)=2 and all leaves at level 2)

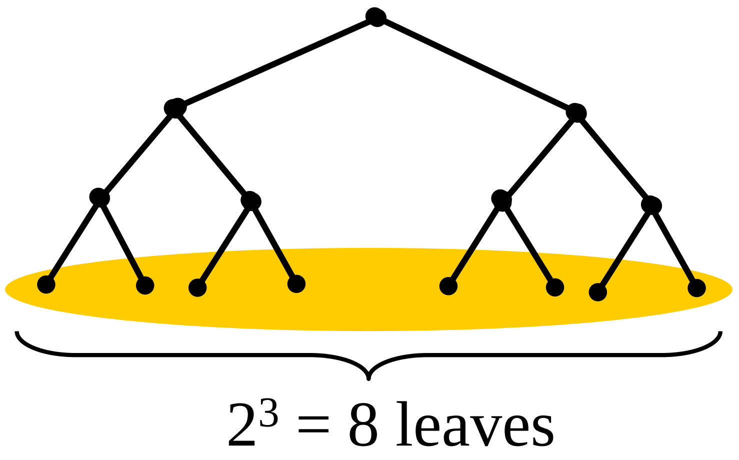
Rooted Trees

Question: How many leaves are there for an m-ary tree of height at most h?

Example: How many leaves are there for a binary tree of height at most 3?

A binary tree (m=2) of height h=3 has the maximum number of leaves, if the tree is full and all leaves are at level 3.

The number of leaves in this case is then $2^3 = m^h$.



Introduction to Trees

Theorem 4: There are at most m^h leaves in an m-ary tree of height h.

Proof: by mathematical induction.

Basic Step (h=1): An m-ary tree of height 1 has at most m^1 leaves.

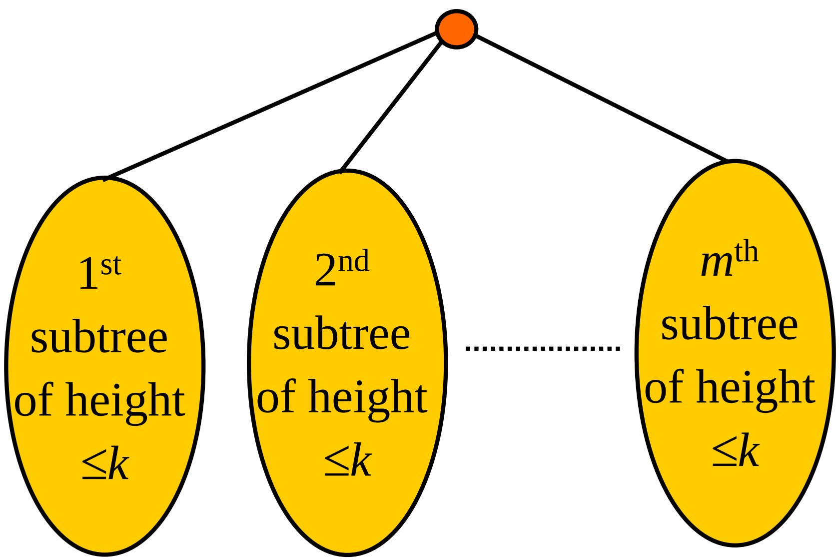
Ind. Hypothesis (h=k): Assume any m-ary tree of height $\le k$ has at most m^k leaves.

Ind. Step (h=k+1): Let T be an m-ary tree of height k+1 (to show: T has at most m^{k+1} leaves)

The leaves of T are the leaves of the subtrees obtained by deleting the edges from the root to each vertex at level 1.

Each of these subtrees has height $\le k$ and therefore at most m^k leaves.

Since there are at most m such subtrees, the maximum number of leaves is $m \cdot m^k = m^{k+1}$.



Introduction to Trees

Let l be the number of leaves of an m-ary tree of height h.

We know that $l \leq m^h$ from Theorem 4. Therefore, $\log_m l \leq h$.

Since h is an integer, we have $\lceil \log_m l \rceil \leq h$

Corollary 1: If an m-ary tree of height h has l leaves, then $h \geq \lceil \log_m l \rceil$

Corollary 2: If an m-ary tree of height h has l leaves and is full and balanced, then $h = \lceil \log_m l \rceil$

Tutorial problems and Additional Reading

- Rosen (8th Edition): 11.1 – Introduction to Trees (page nos 781-793) -- Exercises 11.1 - (48 problems) on pages 791-793.

- Grimaldi (5th Edition): Section 12.1 - Trees – Definitions, Properties, Examples, (page nos 581-587) & part of 12.2 – Rooted Trees – (page nos 587-605). -- Exercises 12.1 (25 problems) pages 585-587; and Exercises 12.2 (25 problems) pages 603-605;

Summary

- Introduction to Trees
- Rooted Trees
- Examples
- References
  - *Rosen* (8th Edition) Section 11.1 – Introduction to Trees (page nos 781-793) ;
  - *Grimaldi* (5th Edition) Section 12.1 - Trees – Definitions, Properties, Examples, (page nos 581-587) & part of 12.2 – Rooted Trees – (page nos 587-605).