

Introduction to Trees

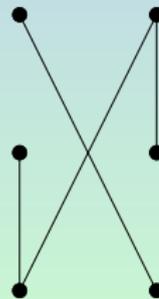
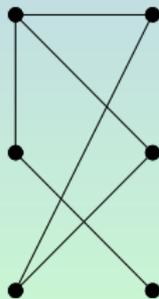
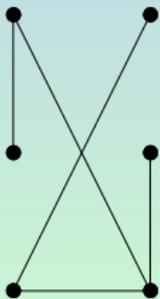
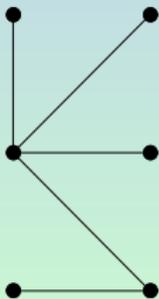
Discrete Mathematics

Definition

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

Because a tree cannot have a simple circuit, a tree cannot contain multiple edges or loops. Therefore any tree must be a simple graph.

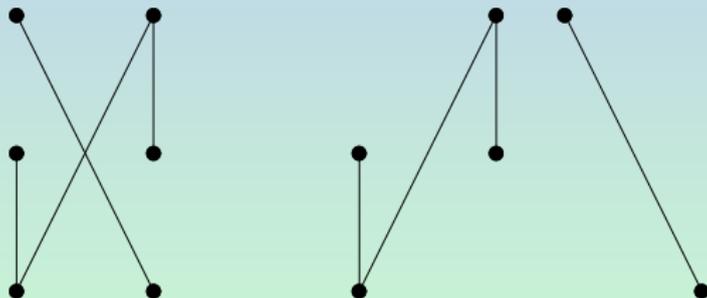
The first two graphs are trees, the last two are not.



Definition

A **forest** is an undirected graph with no simple circuits.

The only difference between a forest and a tree is the word *connected*. If each connected components of the graph is a tree, then the graph is a forest.



Properties of a Tree

Theorem

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

Theorem

A tree is a bipartite graph.

Properties of Trees

Theorem

A tree with n vertices has $n - 1$ edges.

Theorem

If an edge is added to a tree, then a cycle is created.

Trees and Roots

In many applications of trees, a particular vertex of the tree is designated as the **root**.

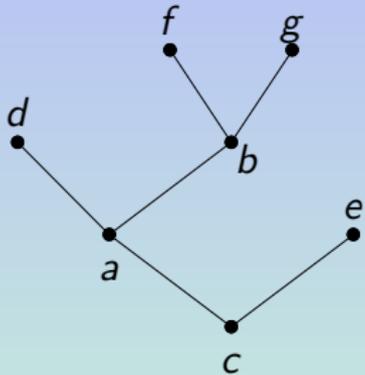
The choice of the root is arbitrary.

Definition

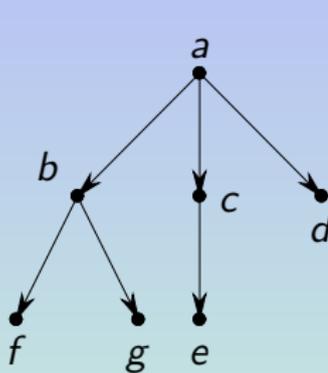
A **rooted tree** is a tree in which one vertex has been designated as the root and every edge is directed away from the root.

We usually draw a rooted tree with its root at the top of the graph. The arrows indicating the directions of the edges can be omitted because the choice of root determines the direction of the edges.

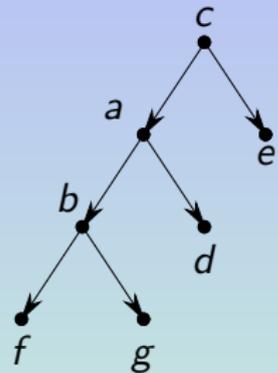
Rooted Tree



Tree



Rooted tree
with a as root

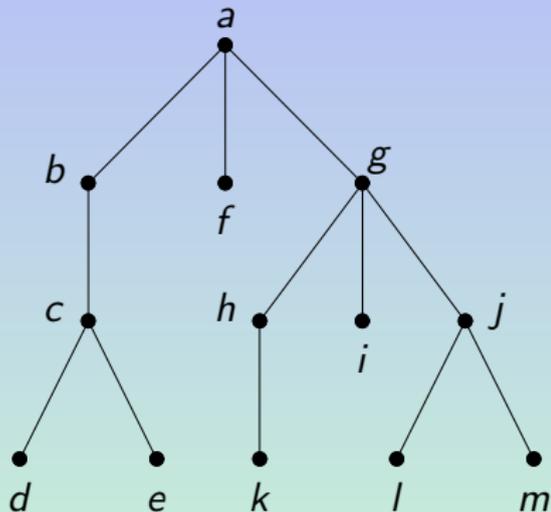


Rooted tree
with c as root

The terminology for trees has botanical and genealogical origins.

- If v is a vertex in 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 , then v is called 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 and including the root.
- The **descendants** of a vertex v are those vertices that have v as an ancestor.
- A vertex of a tree is called **leaf** if it has no children.
- Vertices that have children are called **internal vertices**.

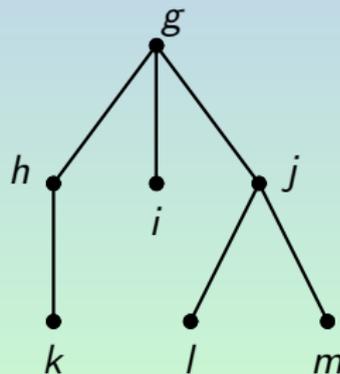
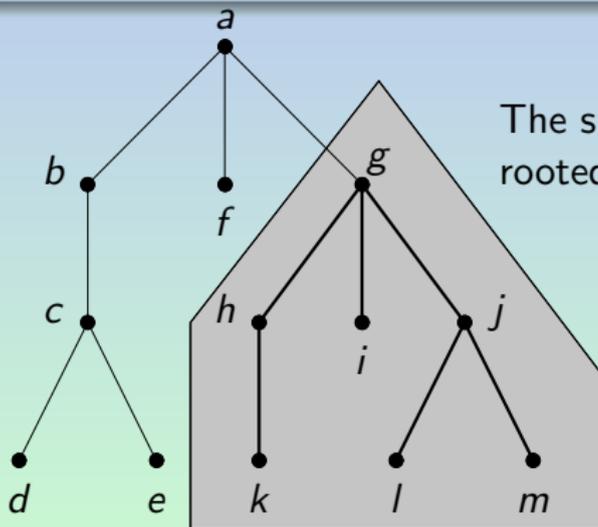
Example of a Rooted Tree



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

Definition

If v is a vertex in a tree, the subtree with v as its root is the subgraph of the tree consisting of v and its descendants and all edges incident to these descendants.

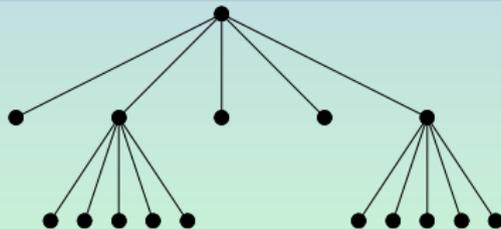


Definition

A rooted tree is called an **m -ary tree** if every internal vertex has no more than m children.

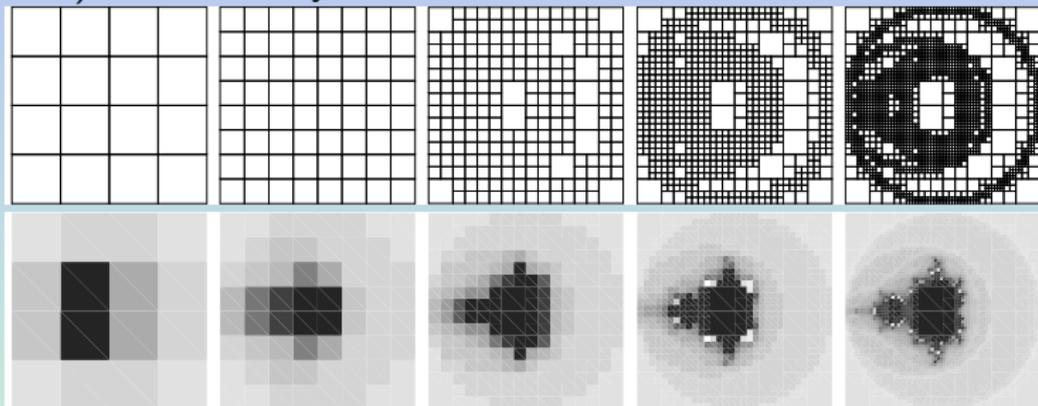
The tree is called a **full m -ary tree** if every internal vertex has exactly m children.

An m -ary tree with $m = 2$ is called a **binary tree**.



Quadtree and Octree

A **quaternary tree (quadtree)** is a full 4-ary tree. An **octary tree (octree)** is a full 8-ary tree.



Mandelbrot fractals on a quadtree of level 2, 3, 4, 5 and 6.

Ordered Rooted Tree

An **ordered rooted tree** is a rooted tree where the children of each internal vertex are ordered. Ordered rooted trees are drawn so that the children of each internal vertex are shown in order from left to right.

In an ordered binary tree, if an internal vertex has two children, the first child is called **left child** and the second child is called **right child**.

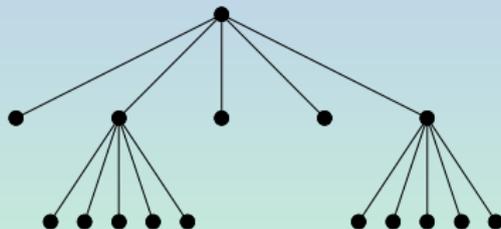
The tree rooted at the left child of a vertex is called the **left subtree** of this vertex, and the tree rooted at the right child of a vertex is called the **right subtree** of this vertex.

Trees as Models

- Family tree
- Molecule representation
- Organizational tree
- File system
- Computer network
- Decision tree
- Sorting tree
- Localization tree
- Etc...

Theorem

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



Properties of Trees

Suppose that T is a full m -ary tree with n vertices. Let i be the number of internal vertices and l the number of leaves in this tree. Once one of n , i , and l is known, the other two quantities are determined.

Theorem

A full m -ary tree with

- i) n vertices has $i = (n - 1)/m$ internal vertices and $l = (n(m - 1) + 1)/m$ leaves,*
- ii) i internal vertices has $n = mi + 1$ vertices and $l = i(m - 1) + 1$ leaves,*
- iii) l leaves has $n = (ml - 1)/(m - 1)$ vertices and $i = (l - 1)/(m - 1)$ internal vertices.*

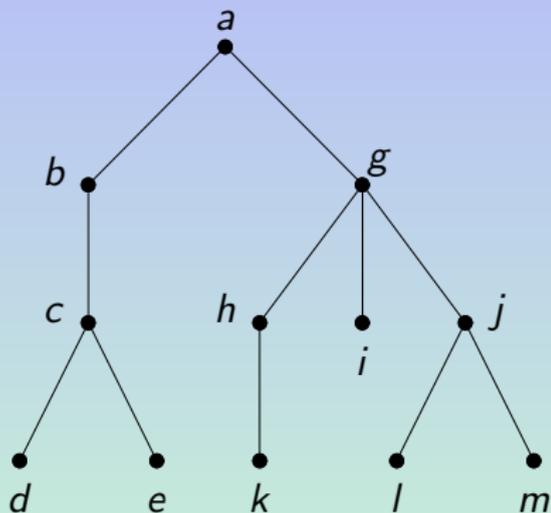
Definition

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

The level of the root is defined to be zero.

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

Example: Level and Height

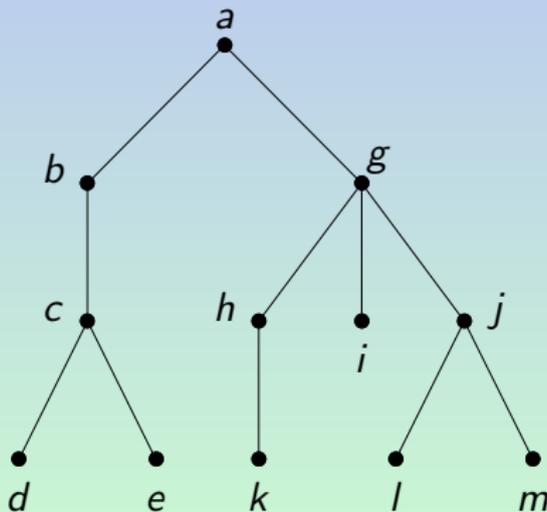


- The root a is at level 0.
- Vertices b and g are at level 1.
- Vertices c , h , i and j are at level 2.
- Vertices d , e , k , l and m are at level 3.
- The height of the rooted tree is 3.

Balanced Tree

Definition

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



Theorem

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

Corollary

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 $h = \lceil \log_m l \rceil$.