Trees

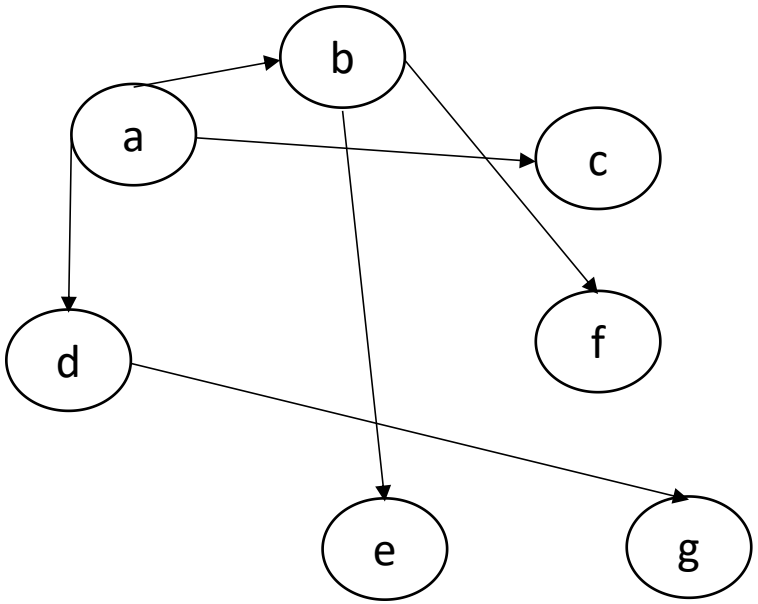
D.E ZEGOUR

Ecole Supérieure d'Informatique

ESI

Trees

E = {a, b, c, d, e, f, g}
Relationships : a R b, a R c, a R d, b R e, b R f, d R g
Graph = { (x, y) in E^2 | x R y }



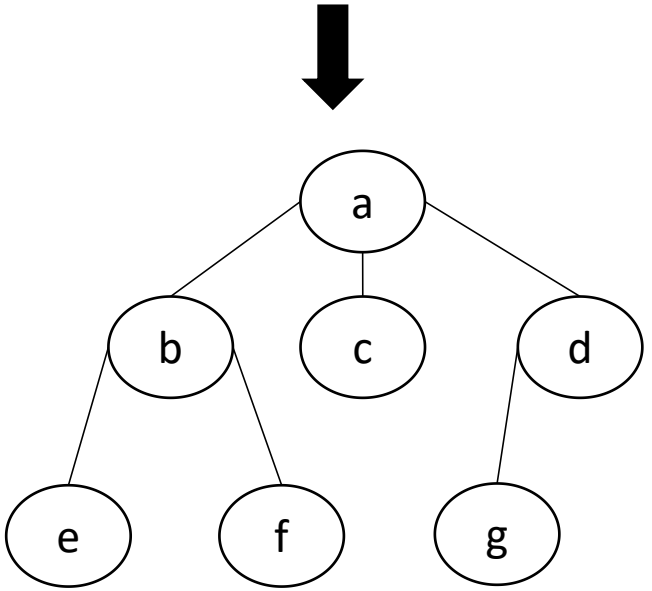
Definition

Hierarchical dynamic data structure

Non-linear linked list of nodes

Mathematically, a tree can be defined by a special binary relation

Each node has at most one predecessor and n successors (n ≥ 0).



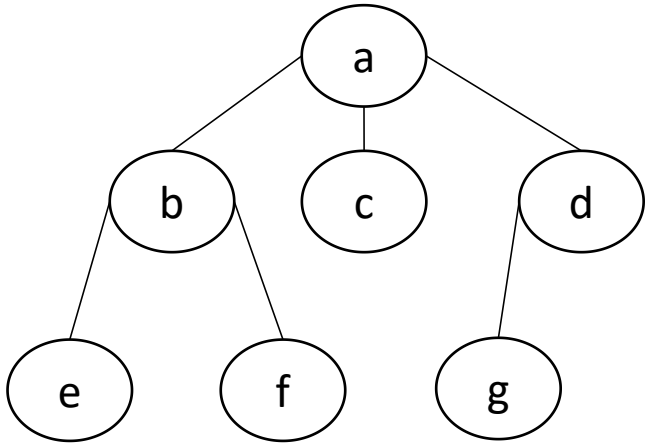
Trees

Definition

Binary tree : number of successors of any node is at most 2

Ternary tree : the number of successors of any node is at most 3

M-ary tree of order n : the number of successors of any node is at most n



Ternary tree

Trees

Examples

family tree :

descendants of a family

A table of contents for a given book :

Book in volumes
Volume in chapters
Chapter in sections
Section in paragraphes
Etc..

Trees

Terminology

Terms derive from the family tree and the natural tree

Root

The root is the node that has no predecessor. In the example, the root is represented by node 'a'.

Leaf (or external node)

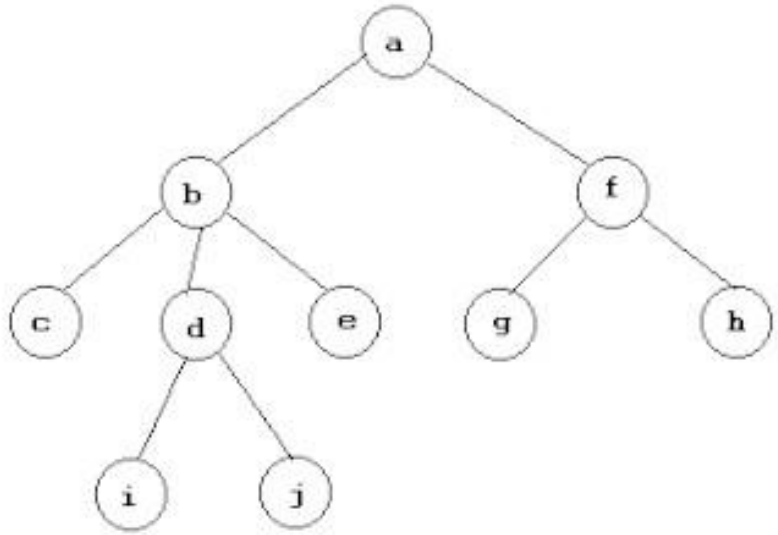
A leaf, or an external node, is a node that has no successor. In the example, the leaves are 'c,' 'i,' 'j,' 'e,' 'g,' and 'h.'

Internal node

An internal node is any node that has at least one successor. In the example, the internal nodes are 'a,' 'b,' 'f,' and 'd.'

Sons (or children) of a node

The sons, or children, of a node are its successors. In the example, 'c,' 'd,' and 'e' are the sons of node 'b.'



Trees

Terminology

Terms derive from the family tree and the natural tree

Brother

Brothers are nodes that share the same parent. In the example, 'c,' 'd,' and 'e' are brothers.

Parent (or Father)

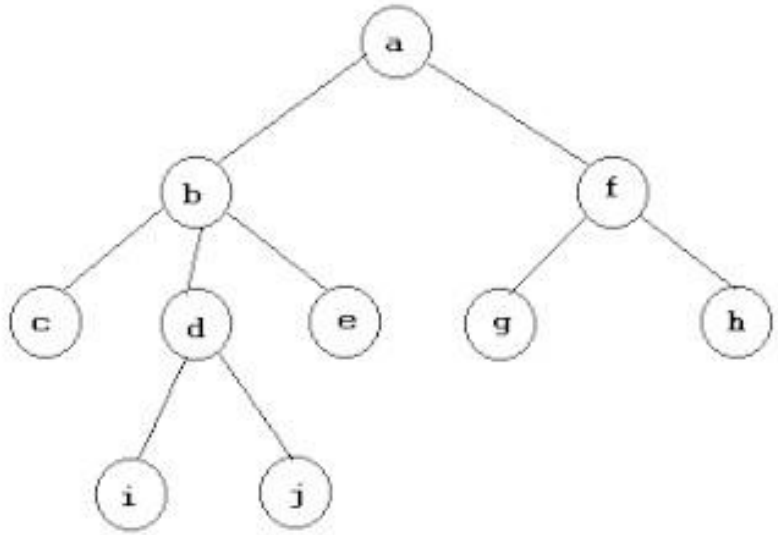
A parent, or father, is a node that has at least one successor. In the example, 'b' is the parent of nodes 'c,' 'd,' and 'e.'

Descendants of a node

Descendants of a node include all the nodes in the subtree rooted at that node. In the example, the descendants of 'h' are 'h,' 'c,' 'd,' 'e,' 'i,' and 'j.'

Ascendants of a node

Ascendants of a node are all the nodes on the branch from the root to that node. In the example, the ascendants of 'd' are 'd,' 'b,' and 'a.' The ascendants of 'g' are 'f' and 'a.'



Trees

Terminology

Terms derive from the family tree and the natural tree

Subtree of root R

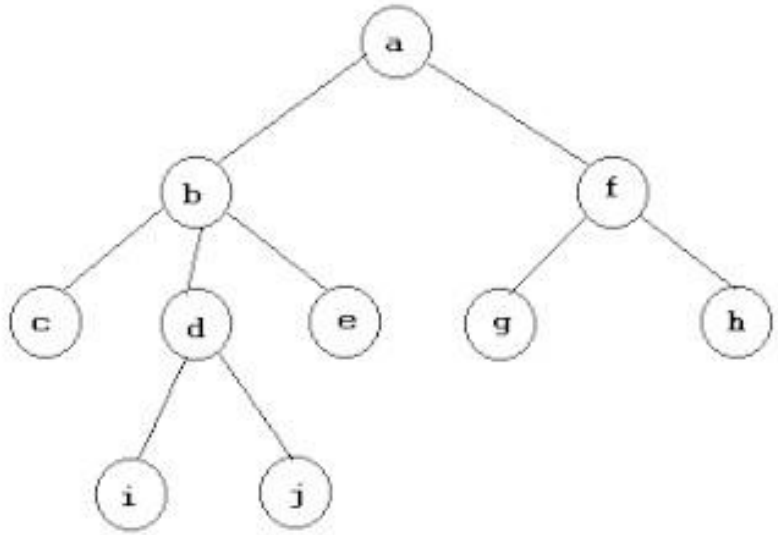
A subtree of root 'R' includes all descendants of 'R.' In the example, node 'f' with its two children 'g' and 'h' constitutes a subtree.

Path

A path consists of the nodes along the branch from the root to a specific node. In the example, 'abd,' 'abdi,' and 'af' are paths.

Branch

A branch consists of the nodes along the path from the root to a leaf. In the example, the branches of the tree are 'a-b-c,' 'a-b-d-i,' 'a-b-d-j,' 'a-b-e,' 'a-f-g,' and 'a-f-h.'



Trees

Terminology

Terms derive from the family tree and the natural tree

Degree of a node

The degree of a node is determined by the number of its children. In the given example, the degree of node 'b' is 3.

Level of a node

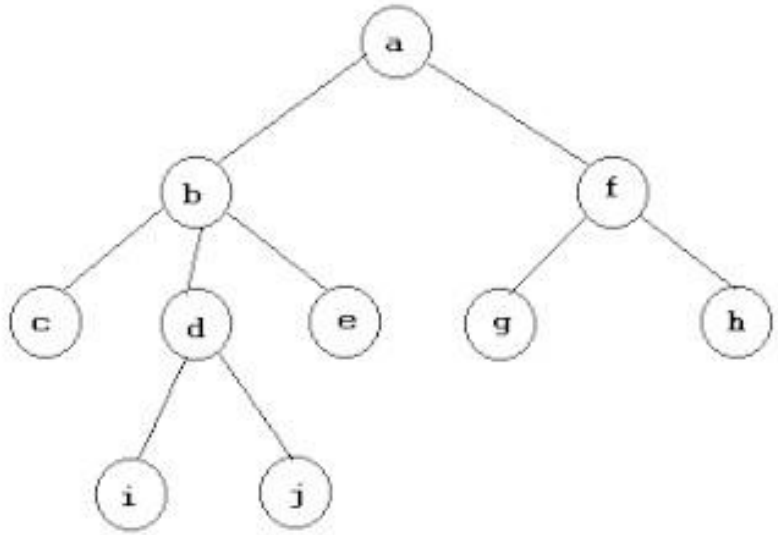
The level of a node is defined as follows: the root is at level 0, its sons are at level 1, their sons are at level 2, and so on. In the example, 'a' is at level 0, 'd' is at level 2, and 'j' is at level 3.

Depth of the tree

The depth of a tree is the maximum level of its leaves. In the example, the tree has a depth of 3.

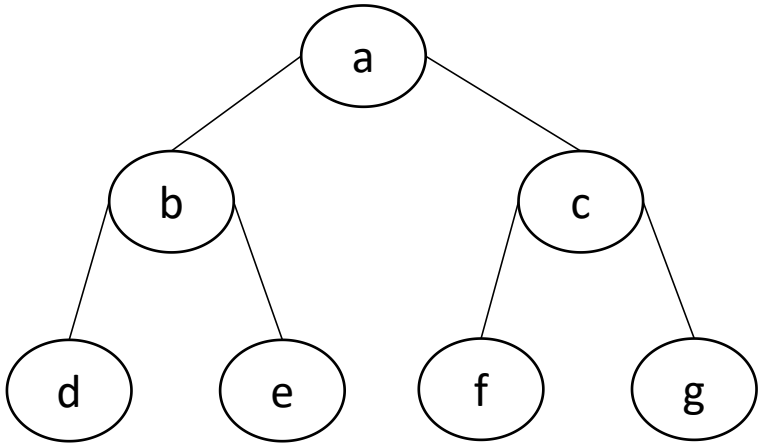
Forest

A forest is a set of trees.



Trees

Binary trees : Terminology / Definitions



Left child, Right child

Left subtree, Right subtree

Strictly binary tree : Every node has exactly 0 or 2 children

Complete binary tree : Strictly binary tree AND all leaves are in the same level.

In a strictly binary tree with n leaves, the total number of nodes is 2n-1

In a complete binary tree of depth n, the total number of nodes is $2^{d+1} - 1$

Trees

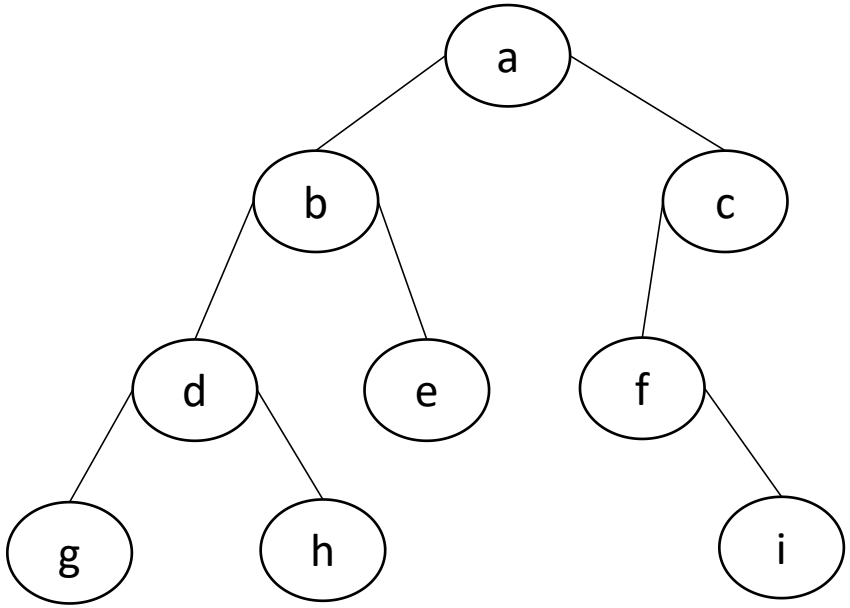
Abstract machine

LC (P) : Access to field Left Child of the node referenced by P.
RC (P) : Access to field Right Child of the node referenced by P.
PARENT (P) : Access to field Parent of the node referenced by P.
NODE_VALUE (P) : Access to field Value of the node referenced by P.

ALLOCATE_NODE (Val) :
Create a node with value Val and return the address of the node. The others fields are to Nil.
FREE_NODE (P) :
Free the node at address P.

ASS_LC (P, Q) : Assign the address Q to the field Left Child of the node referenced by P.
ASS_RC (P, Q) : Assign the address Q to the field Right Child of the node referenced by P.
ASS_PARENT (P, Q) : Assign the address Q to the field parent of the node referenced by P.
ASS_NODE_VAL(P, Val) : Assign the value Val to the field Value of the node referenced by P.

Trees



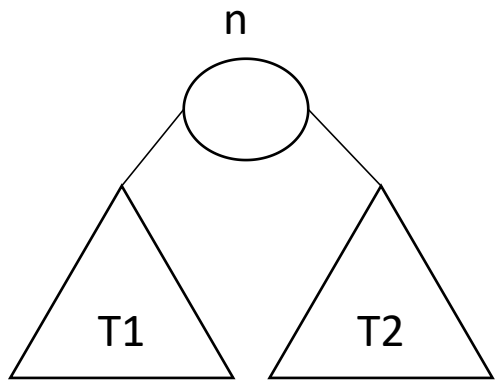
Traversal

There are more than thirty traversal orders.

The most commonly used:

Preorder :	n T1 T2
Inorder :	T1 n T2
Postorder :	T1 T2 n

Reverse Preorder :	n T2 T1
Reverse Inorder:	T2 n T1
Reverse Postorder:	T2 T1 n



Breadth traversal : Level by level, Left to Right
Breadth traversal : Level by level, Right to Left

Trees

Traversal : Examples

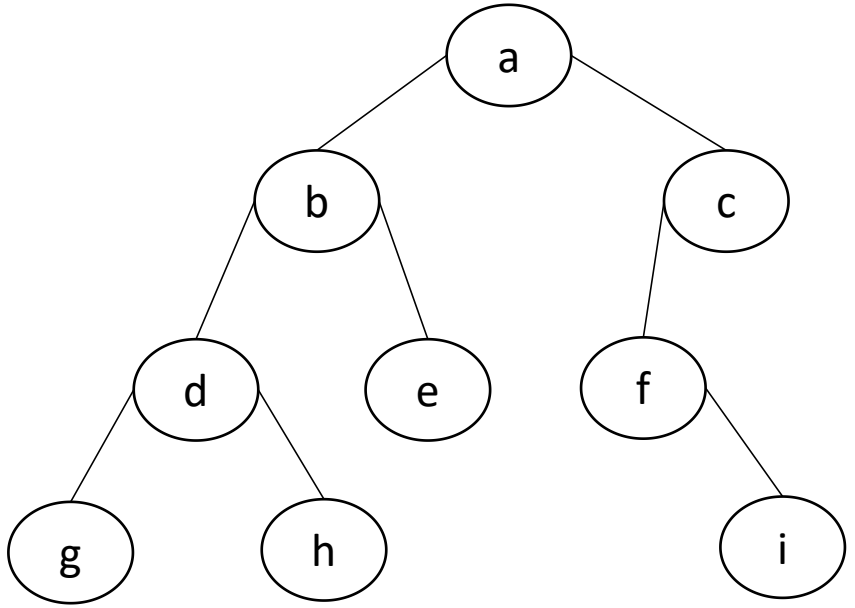
Preorder (nT1T2): a, b, d, g, h, e, c, f, i

Inorder(T1nT2) : g, d, h, b, e, a, f, i, c

Postorder (T1T2n) : g, h, d, e, b, i, f, c, a

Reverse Preorder (nT2T1) : a, c, f, i, b, e, d, h, g

Breadth (Left to Right) : a, b, c, d, e, f, g, h, i



Trees

Traversal : Algorithms

Preorder
Formula : n T1 T2

```
Preorder(n)
IF n <> Nil
    Write(Node_value(n))
    Preorder (LC(n))
    Preorder (RC(n))
ENDIF
```

Preorder (n)
it's possible to write an iterative algorithm using a stack

Breadth (Left to Right)

```
Createqueue(Q)
Enqueue(Q, r)
WHILE not Empty_queue(Q)
    Dequeue(Q, n)
    Write(Node_value(n))
    IF Lc(n) <> Nil Enqueue(Q, LC(n))   ENDIF
    IF Rc(n) <> Nil Enqueue(Q, RC(n))  ENDIF
ENDWHILE
```