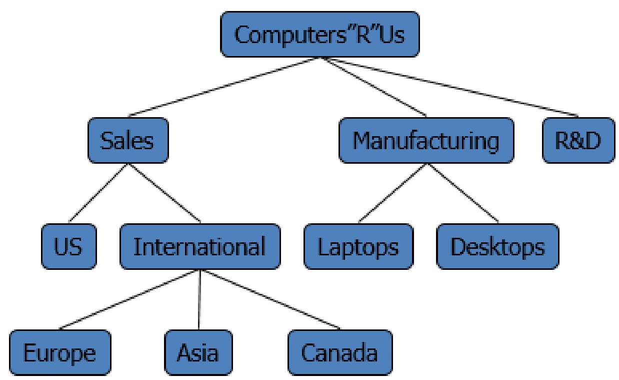
Chapter 15 -1 Trees - Terminology

Data Structure

- A **linear data structure** is one in which, while traversing sequentially, we can reach only one element directly from another
  - Linked list, array
- In a **nonlinear data structure**, the components do not form a simple sequence of first entry, second entry, third entry, and so on.
  - Instead, there is a more complex linking between the components
  - Trees

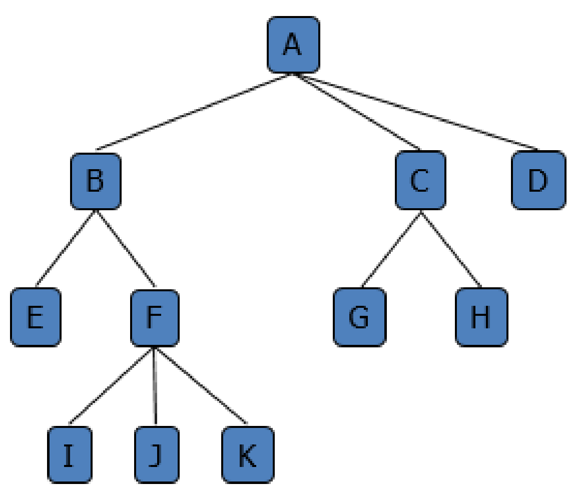
What is a Tree

- In computer science, a tree is an abstract model of a hierarchical structure
- A tree consists of nodes with a parent-child relation
- Applications:
  - Organization charts
  - File systems
  - Programming environments



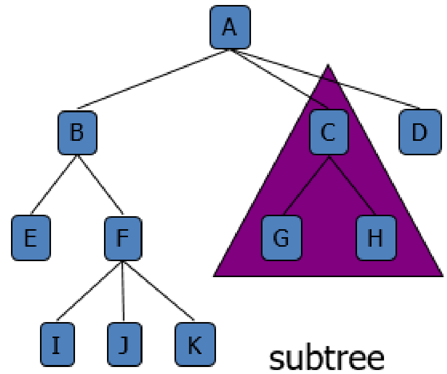
Tree Terminology

- **Node**: simple object that can have a name and can carry other associated information; node A, B, C, D, etc.
- **Edge**: connection between two nodes; edge (A, B), edge (B, E), etc.
- If an edge is between node N and M, and node N is above node M in the tree, then N is the **parent** of M and M is a **child** of N; node B is the parent of node E & F, node I, J, & K are children of node F
- **Siblings**: children of the same parent; node I, J, & K are siblings
- **Root**: the first or top node in a tree; node without parent (A); all nodes except root have exactly one parent



- **Path**: list of distinct nodes in which successive nodes are connected by path in the tree; {A, B, E}, {A, B, F, J}, {A, C, G}, etc.
- **Internal node**: node with at least one child; A, B, C, F
- **External node** (a.k.a. **leaf**): node without children; E, I, J, K, G, H, D
- **Ancestors** of a node: parent, grandparent, grand-grandparent, etc.; node A is an ancestor of node F, node B is an ancestor of node J

- **Depth** (or **Level**) of a node: number of ancestors; depth of node A is 1, depth of node B is 2, depth of node K is 4
- **Height** of a tree: maximum depth of any node; height is 4
- **Descendant** of a node: child, grandchild, grand-grandchild, etc.; node E, F, I, J, & K are descendants of node B
- **Subtree**: tree consisting of a node and its descendants; subtree of rooted at node C
- **M-ary tree**: each node has at most M children; 3-ary tree, 4-ary tree, binary tree



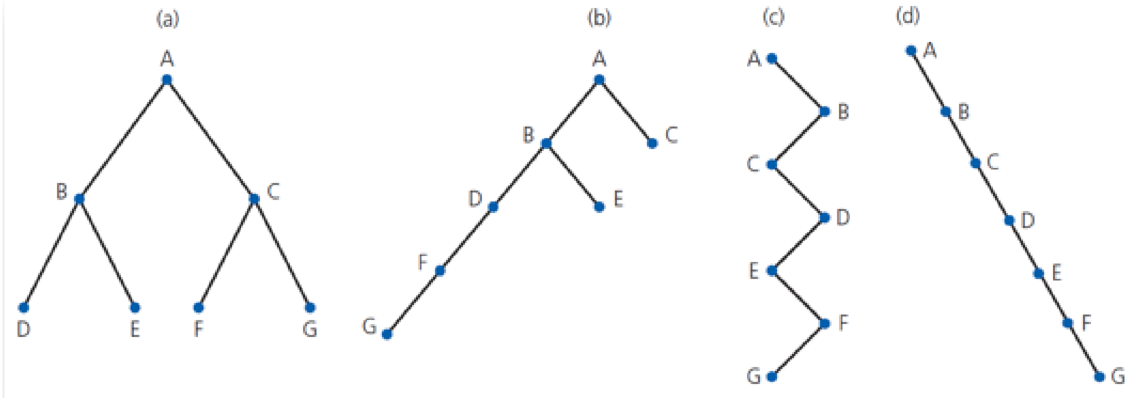
<u>Kinds of Trees</u>

- General tree
  - Set *T* of one or more nodes
  - *T* is partitioned into disjoint subsets
- Binary tree
  - Set of *T* nodes – either empty or partitioned into disjoint subsets
  - Single node *r*, the root
  - Two (possibly empty) sets – left and right subtrees

<u>The height of Trees</u>

- Level of a node, *n*
  - If *n* is root, level 1
  - If *n* not the root, level is 1 greater than level of its parent
- Height of a tree
  - Number of nodes on longest path from root to a leaf
  - *T* empty, height 0
  - *T* not empty, height equal to max level of nodes

Binary trees with the same nodes but different heights



- A recursive definition of height
  - If T is empty, its height is 0
  - If T is not empty,

$$height(T) = 1 + \max\{height(T_L), height(T_R)\}$$

$$\begin{array}{ccc} & r & \\ / & & \backslash \\ T_L & & T_R \end{array}$$

<u>Full Binary Trees</u>

- A binary tree of height h is ***full*** if
  - Nodes at levels $< h$ have two children each
- Recursive definition
  - 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 binary trees of height $h - 1$

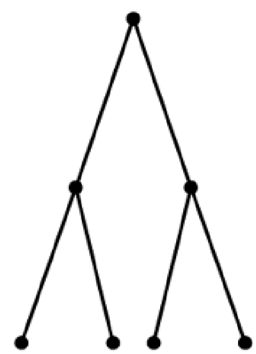
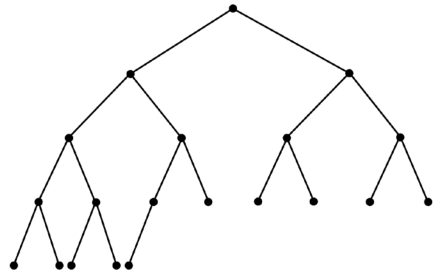


Figure 10-7

A full binary tree of height 3

Complete Binary Trees

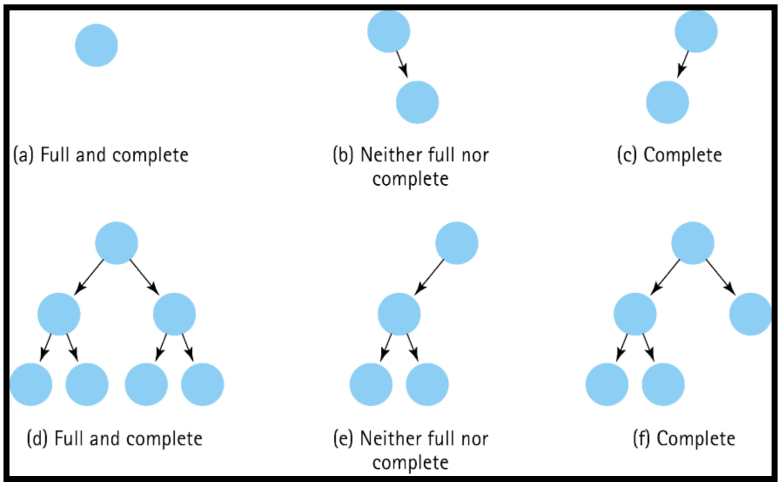
A binary tree of height h is ***complete*** if
It is full to level $h - 1$, and level h is filled from left to right



Balanced Trees

- A binary tree is ***balanced*** if the heights of any node's two subtrees differ by no more than 1
- Complete binary trees are balanced
- Full binary trees are complete and balanced

Examples of Different Types of Binary Trees



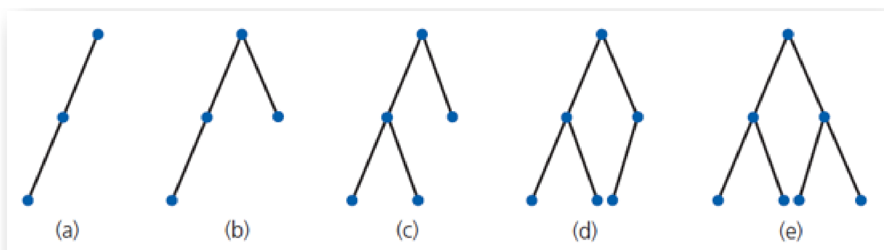
Binary Trees

- A binary tree is a tree with the following properties:
  - Each internal node has up to two children
- We call the children of an internal node left child and right child
- Alternative recursive definition: a binary tree is either
  - a tree consisting of a single node, or
  - a tree whose root has an ordered pair of children, each of which is a binary tree

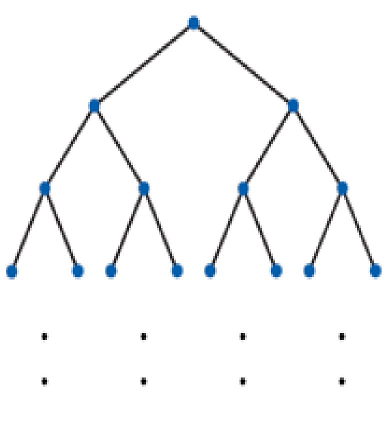
The maximum and minimum heights of a binary tree

- Binary tree with *n* nodes
  - Max height is *n*
- To minimize height of binary tree of *n* nodes
  - Fill each level of tree as completely as possible
  - A complete tree meets this requirement

Binary Trees of height 3



Counting the nodes in a full binary tree of height *h*



Level	Number of nodes at this level	Total number of nodes at this level and all previous levels
1	$1 = 2^0$	$1 = 2^1 - 1$
2	$2 = 2^1$	$3 = 2^2 - 1$
3	$4 = 2^2$	$7 = 2^3 - 1$
4	$8 = 2^3$	$15 = 2^4 - 1$
⋮	⋮	⋮
h	2^{h-1}	$2^h - 1$