National University of Computer & Emerging Sciences

Trees

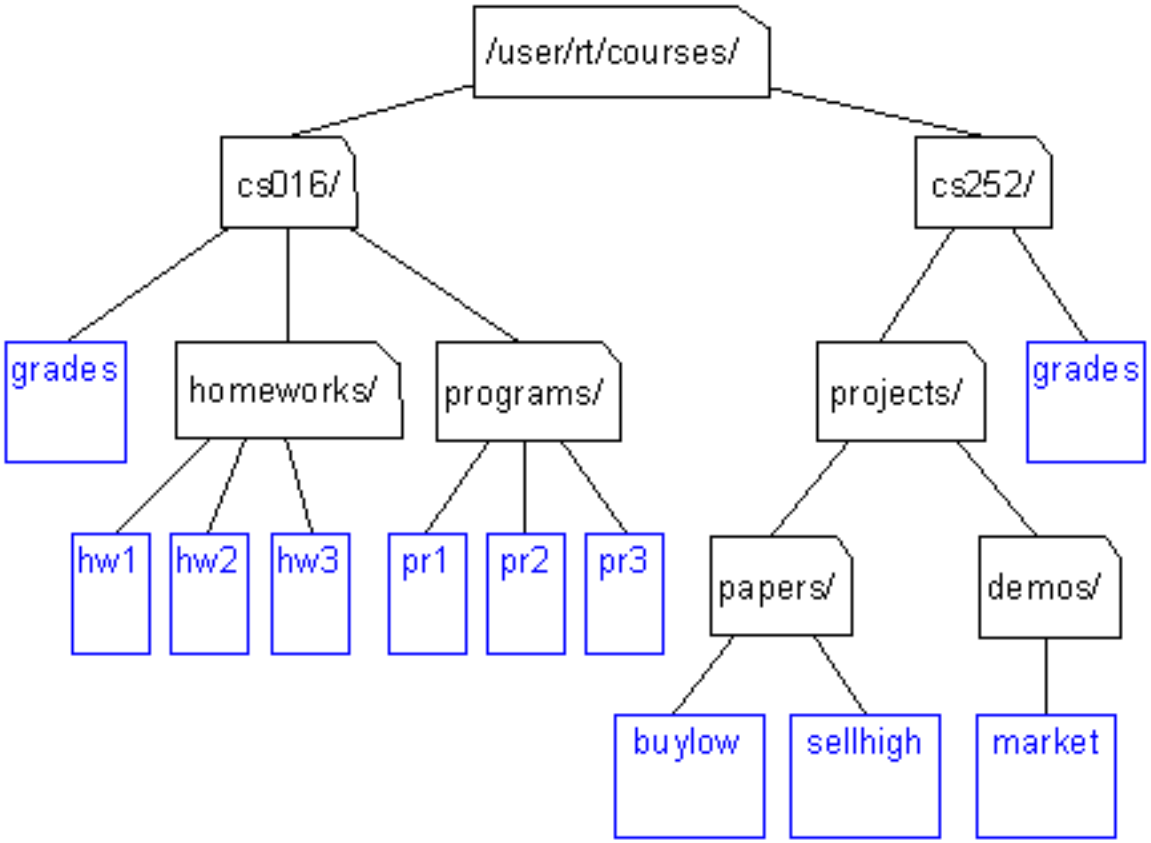
Trees

- *Hierarchical* data structure

- Examples:
  - TOC in a book have a shallow tree structure
  - A family tree

  - Others?

Unix / Windows file structure



Trees: Basic terminology[1]

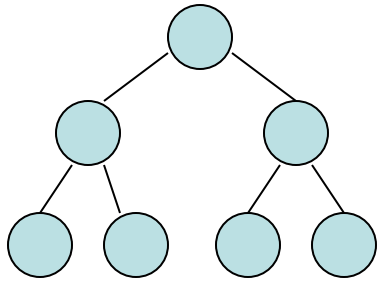
- *Hierarchical* data structure
- Each position in the tree is called a *node*
- The “top” of the tree is called the ***root***

- The nodes immediately below a node are called its ***children***; nodes with no children are called ***leaves (or terminal nodes)***, and the node above a given node is its ***parent (or father)***

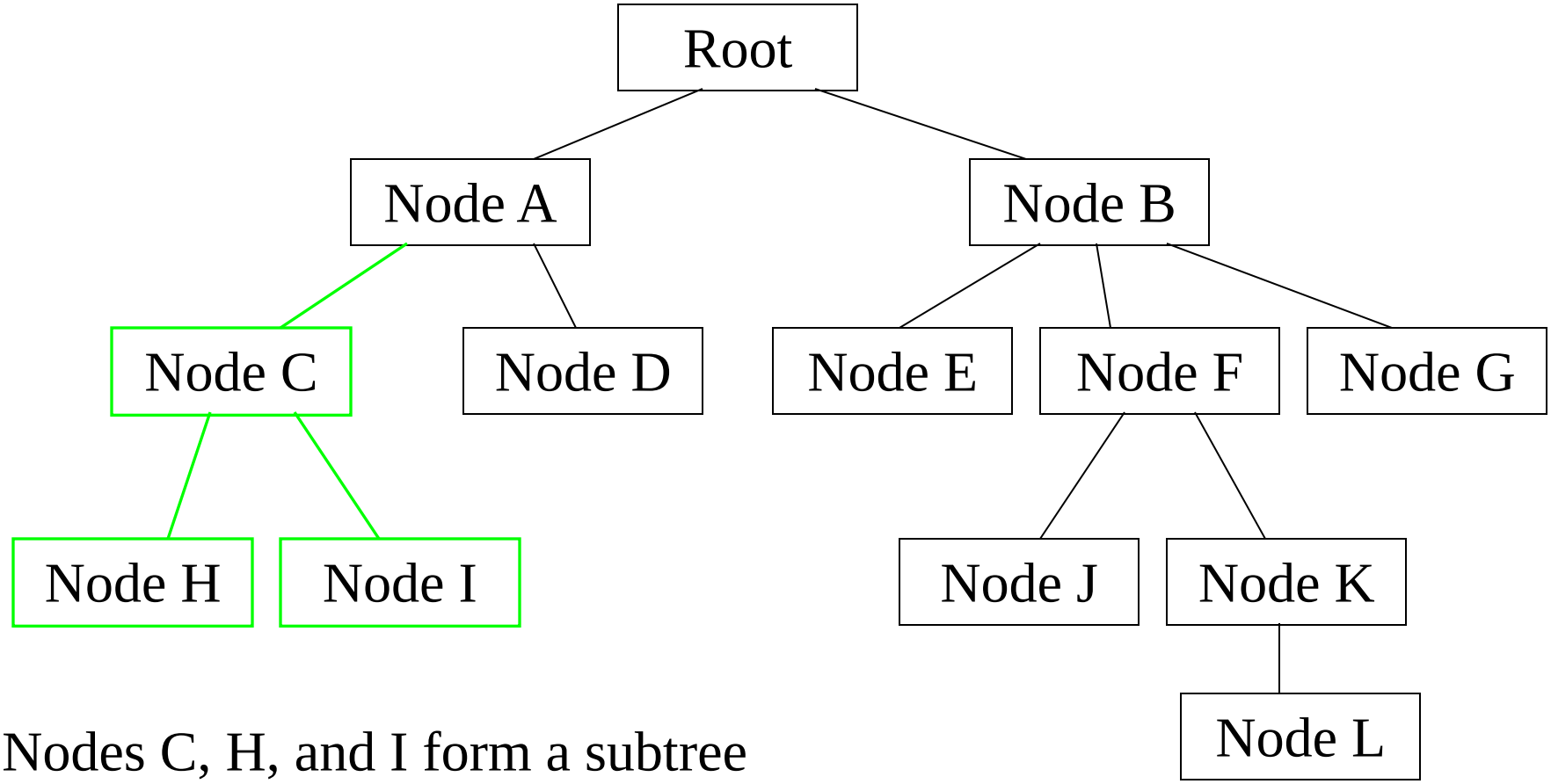
- A node x is ***ancestor*** of node y if x is father of y or father of some ancestor of y. y is called ***descendent*** of x. Ancestor of a node is its parent, grand parent or grand-grand parent or so on….

Trees: Basic terminology[2]

- Nodes with the same parent are ***siblings***
- A node and collection of nodes beneath it is called a ***subtree***
- The number of nodes in the longest path from the root to a leaf is the ***depth*** (or ***height***) of the tree
  - is the depth 2 or 3?
  - depends on the author



- Text book definition of *depth* of a tree
  - Depth of a tree is maximum level of any leaf in the tree.
  - Root of tree is at level 0, and level of any other node in the tree is one more than the level of its father



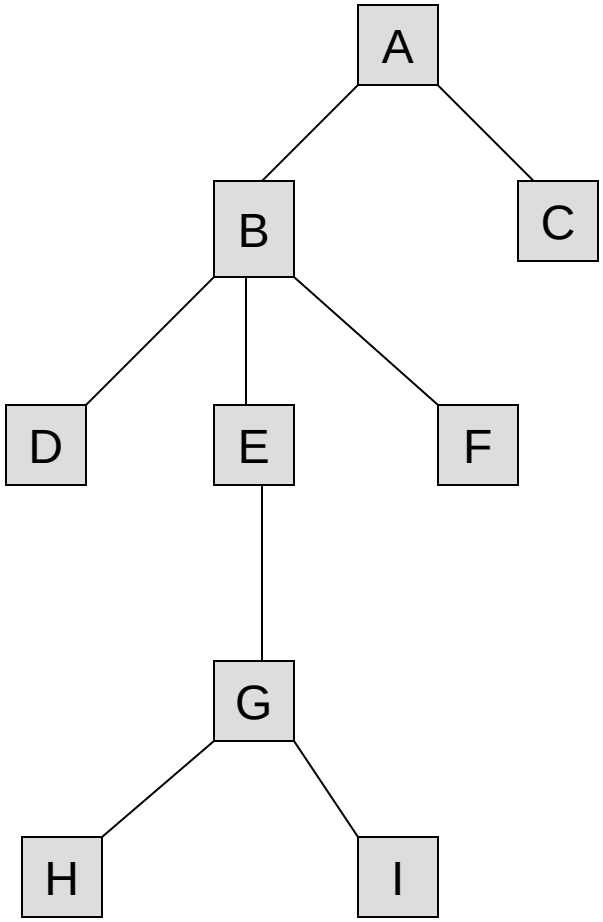
Nodes C, H, and I form a subtree

Nodes H and I are siblings

C is H's parent, H and I are children of C

What is the depth of this tree?

Tree Properties



Property	Value
Number of nodes	
Height	
Root Node	
Leaves	
Ancestors of H	
Descendants of B	
Siblings of E	
Right subtree	

BINARY TREES

Binary Trees

- A commonly used type of tree is a binary tree
- Each node has *at most* two (bi) children

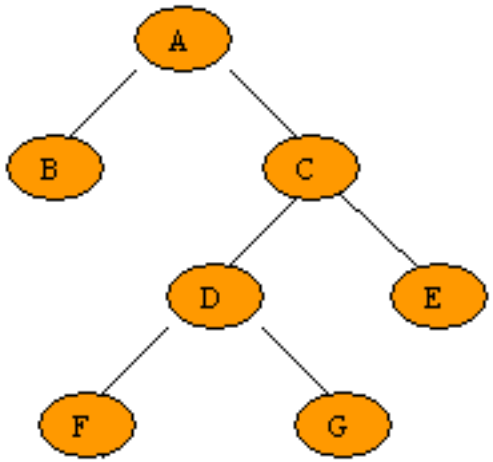
- The “tree” is a conceptual structure
- The data can be stored either in
  - a dynamic linked tree structure, or
  - in contiguous memory cells (array) according to a *set pattern*;

- In other words, implementation can be pointer-based or array-based

Binary Trees -Types

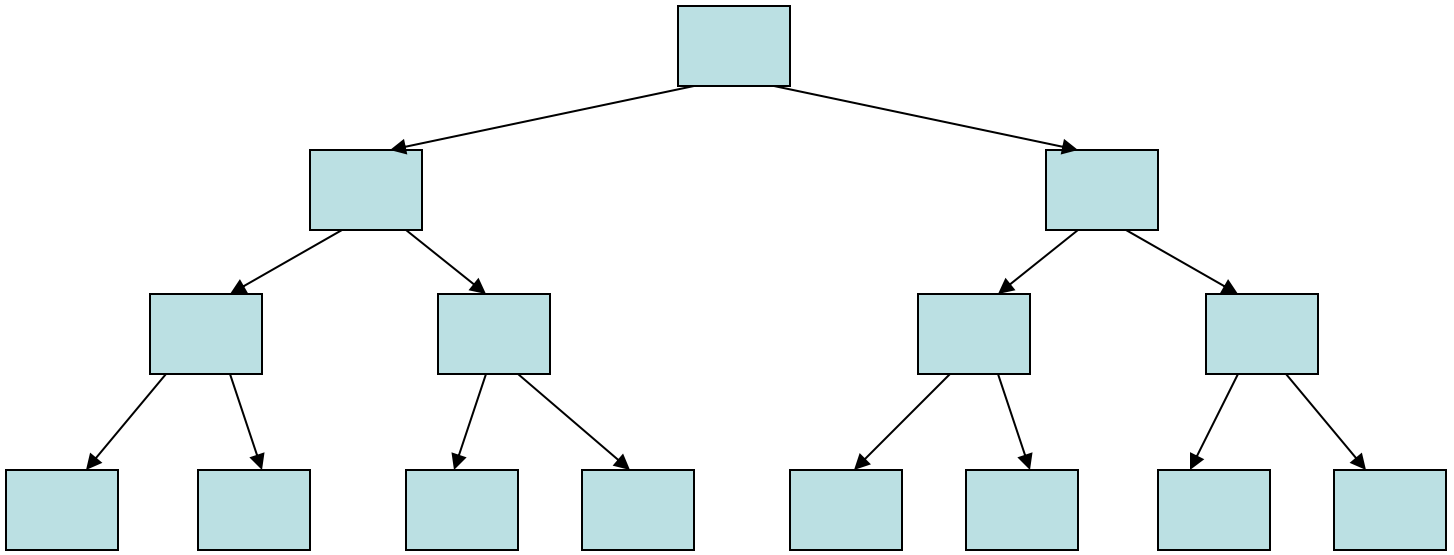
- A **rooted binary tree** is a tree with a root node in which every node has at most two children. There are also un-rooted/free trees (known as graphs)!

- A **full binary tree** (sometimes **proper binary tree** or **2-tree** or **strictly binary tree**) is a tree in which every node other than the leaves has two children. Or, perhaps more clearly, every node in a binary tree has exactly (strictly) 0 or 2 children.



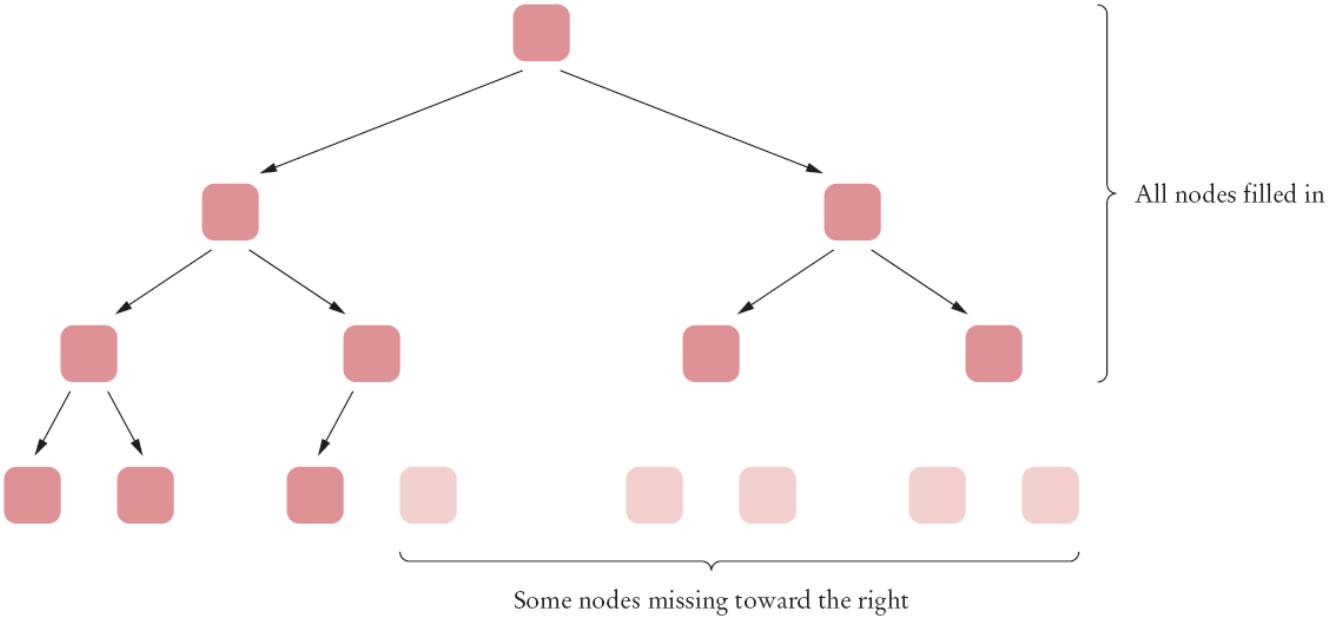
Binary Trees -Types

- A **complete (or perfect) binary tree** is a binary tree in which every level is completely filled.
  - Example?
  - A family tree?



Binary Trees -Types

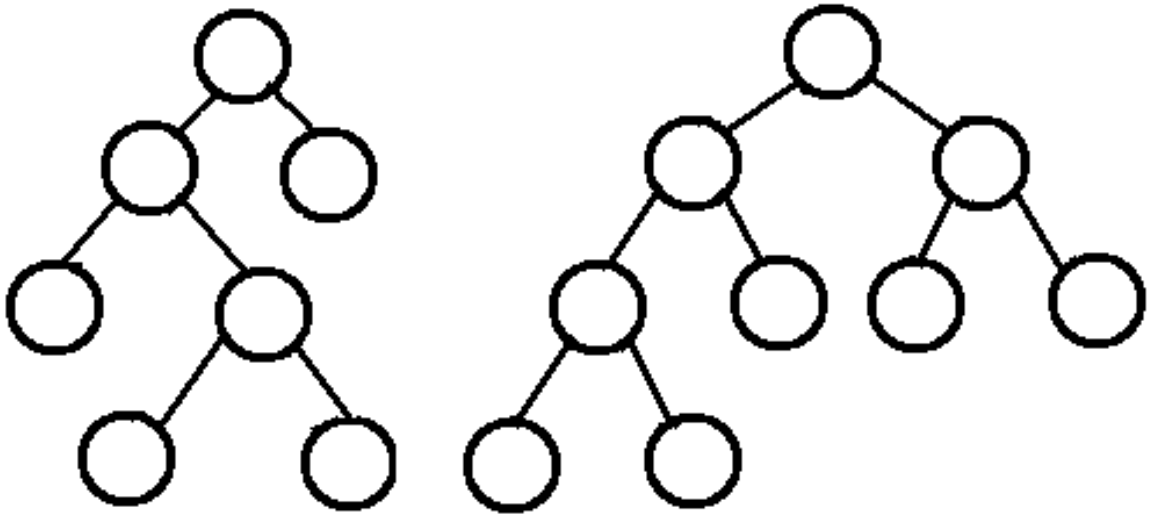
- A tree is called an **almost complete binary tree** or nearly complete binary tree if the last level is not completely filled and all nodes are as far left as possible.



An Almost Complete Tree

Binary Trees -Types

- What is the difference between full and complete binary tree?

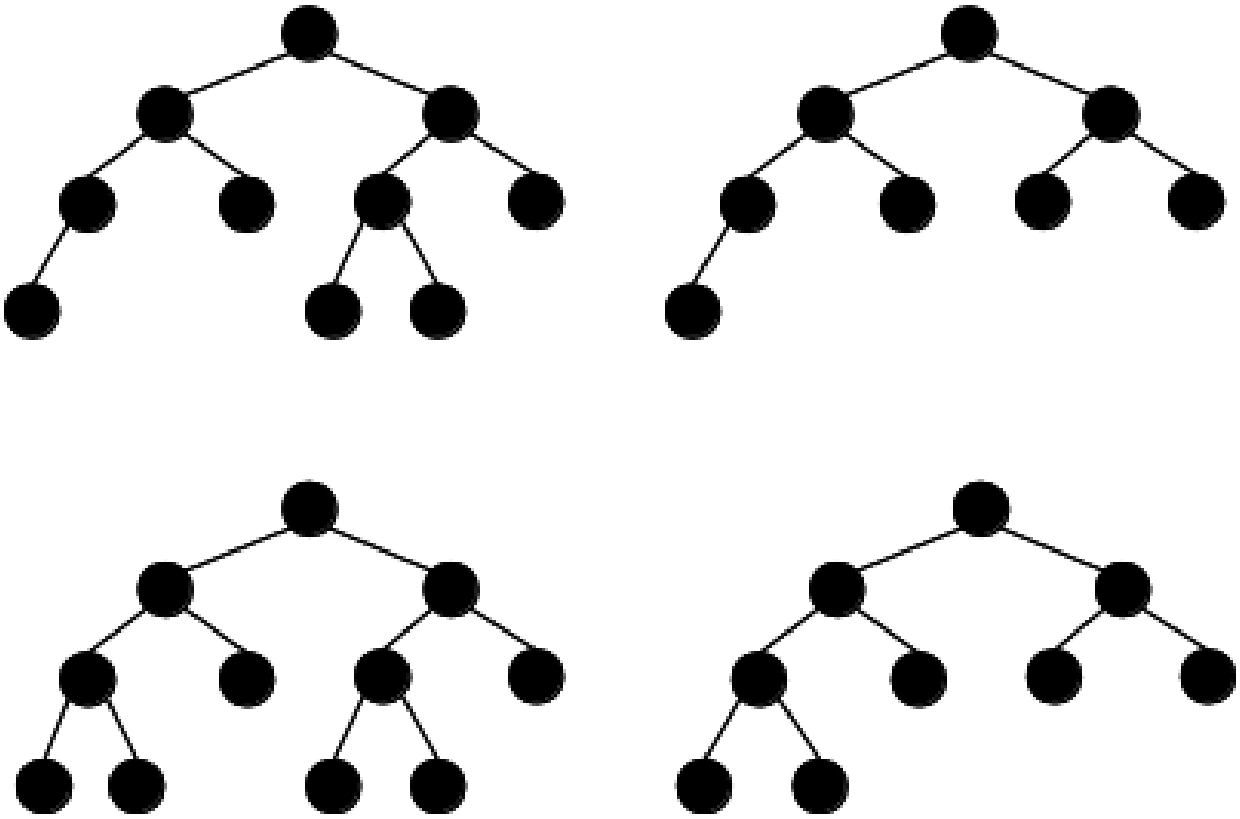


full tree

complete tree

Binary Trees -Types

- What is the difference between full and complete binary tree?



Questions?

"He who asks a question is a fool for five minutes; he who does not ask a question remains a fool forever"

Chinese Proverb

"The wise man doesn't give the right answers, he poses the right questions."

Claude Levi-Strauss

"A wise man can learn more from a foolish question than a fool can learn from a wise answer."

Bruce Lee