Binary Trees

Dr. Nguyen Hua Phung

HCMC University of Technology, Viet Nam

01, 2020

Outline

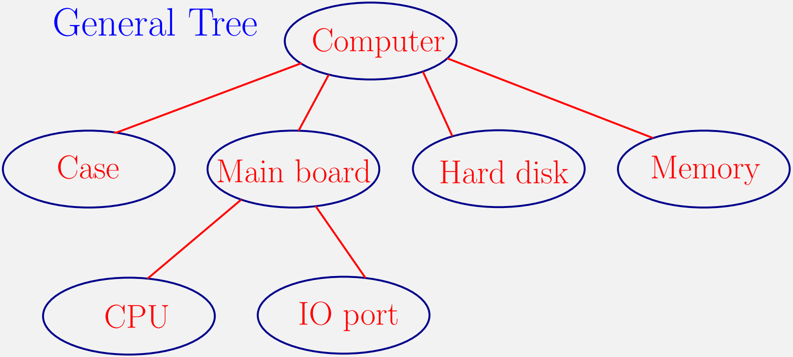
Basic Tree Concepts

Binary Trees

Basic Tree Concepts

- A tree consists of
    - nodes: finite set of elements
    - branches: directed lines connecting the nodes
- For a node:
    - degree: number of branches associated with the node
    - indegree: number of branches towards the node
    - outdegree: number of branches away from the node
- For a tree:
    - root: node with indegree 0
    - nodes different from the root must have indegree 1

Tree Representation



Terminology

- Leaf: node with outdegree 0
- Internal node: not a root or a leaf
- Parent: node with outdegree greater than 0
- Child: node with indegree greater than 0
- Siblings: node with the same parent
- Path: sequence of adjacent node

Terminology

- Ancestor: node in the path from the root to the node
- Descendent: not in a path from the node to a leaf
- Level: the node's distance from the root (at level 0)
- Height (Depth): the level of the leaf in the longest path from the root plus 1
- Sub-tree: connected structure below the root

Tree representation

- Indented list

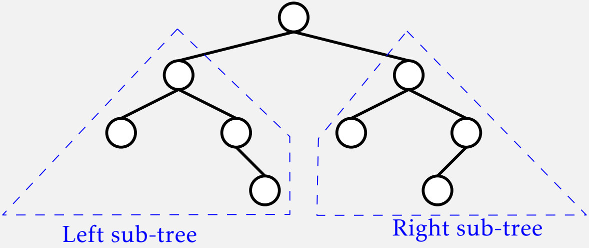
  Computer
  - Case
  - Main board
    - CPU
    - IO port
  - Hard disk
  - Memory

- Parenthical listing

  Computer(Case, Main board(CPU, IO Port), Hard disk, Memory)

Binary Trees

- A binary tree is a tree in which each node has at most two sub-trees



Binary Tree Properties

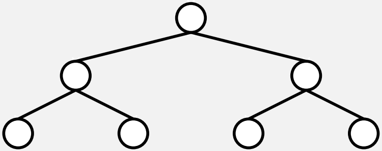
- Height (H) of binary trees vs. Number (N) of nodes:

$$\begin{aligned} \mathrm{H}_{max} &= \mathrm{N} \\ \mathrm{H}_{min} &= \lfloor log_2 N \rfloor + 1 \\ \mathrm{N}_{min} &= \mathrm{H} \\ \mathrm{N}_{max} &= 2^H - 1 \end{aligned}$$

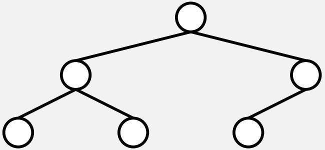
- Balance:
  - Balance factor: $\mathrm{B} = \mathrm{H}_L - \mathrm{H}_R$
  - Balanced tree:
    - |B| is 0 or 1 and
    - all sub-trees are balanced

Binary Tree Properties

- Completeness:



- Complete tree:
$N = N_{max} = 2^H - 1$
(last level is full)



- Nearly complete tree:
$H = H_{min} = \lfloor log_2 N \rfloor + 1$
nodes in the last level are on the left

Binary Tree Abstract Data Type (ADT)

A Binary Tree ADT is either empty, or a node called root which contains two binary trees called the left and the right subtree of the root.

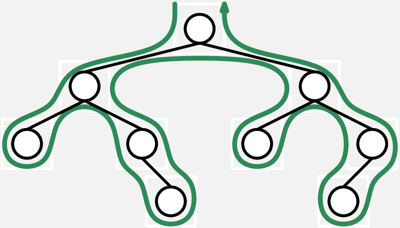
Basic operations:

- *Construct* a tree, leaving it empty
- *isEmpty* to check it empty or not
- *isComplete* to check it complete or not
- *size* to return the number of nodes
- *Insert* an element.
- *Remove* an element.
- *Search* an element.
- *Retrieve* an element.
- *Traverse* the tree, performing a given operation on each element.

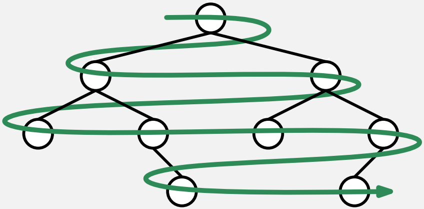
Binary Tree Traversal

Each node is processed once and only once in a predetermined sequence.

- Depth-First Traversal:
  - Node-Left-Right (NLR) or Pre-Order Traversal
  - Left-Node-Right (LNR) or In-Order Traversal
  - Left-Right-Node (LRN) or Post-Order Traversal
- Breadth-First Traversal



Depth-FirstTraversal



Breadth-First Traversal