Lecture 15

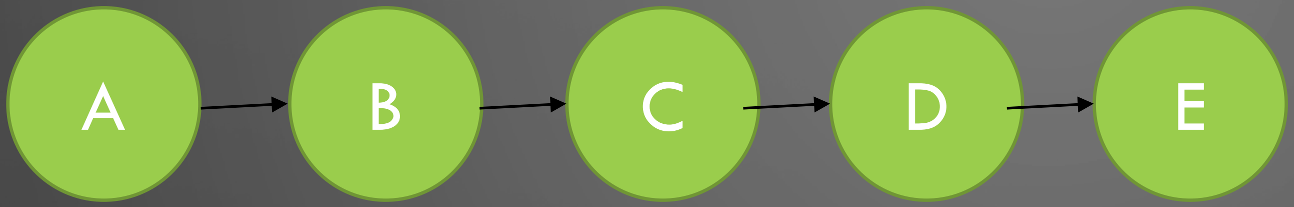
GENERAL AND BINARY TREES

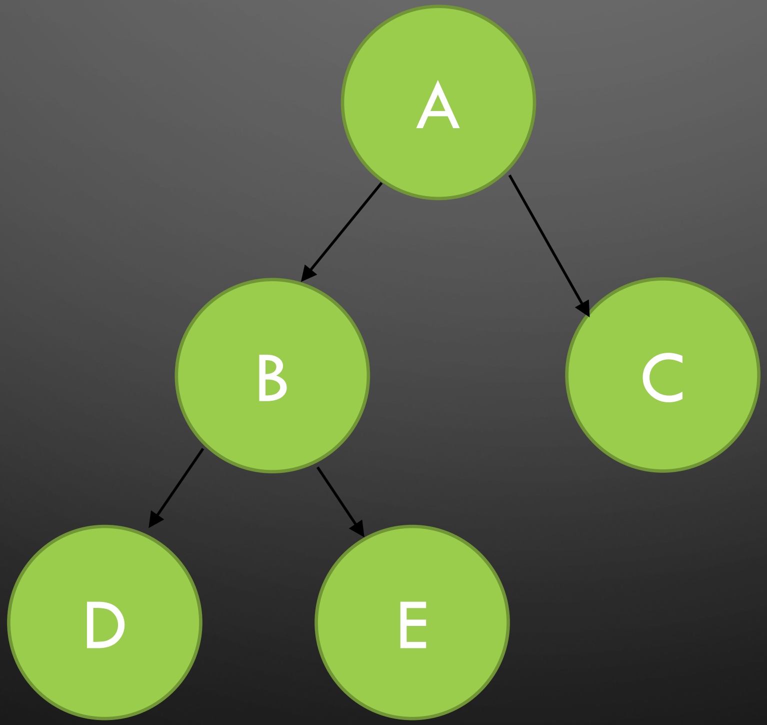
Outline

- **What are Trees:**
  - **Trees Terminology**
  - **Binary Trees**
- Examples

What are Trees?

List: Implicit Linear Order

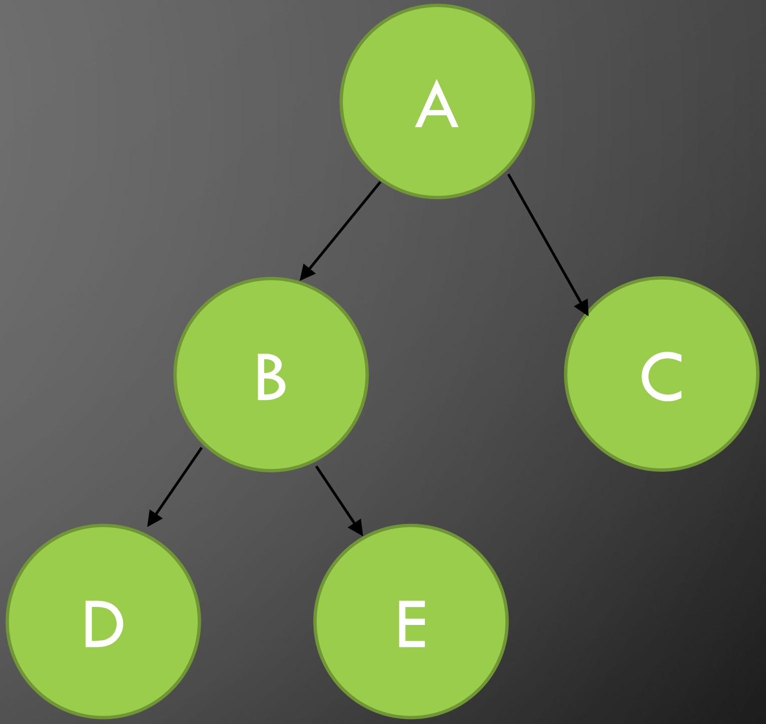




Tree: Implicit Hierarchical Order

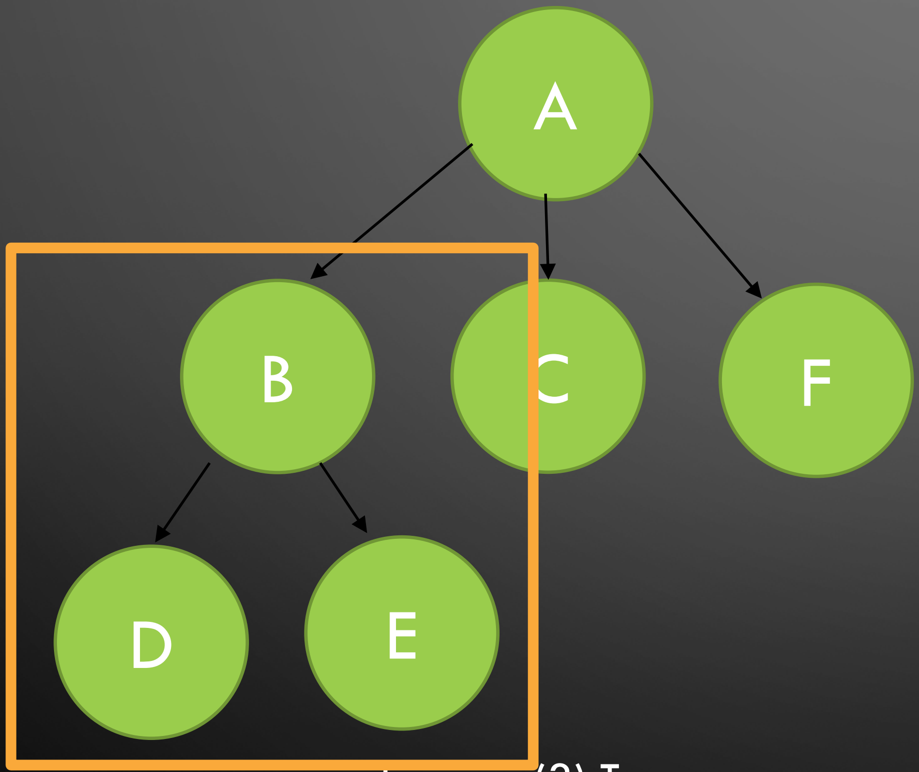
Terminology: Within a Tree

- *Node (vertex)* [A,B,C,D,E]
- *Edges (links)* [A-B, A-C, B-D, B-E]
- *Parent* [A parent of B, C; B parent of D, E]
- *Child* [B, C child of A; D, E child of B]
- *Sibling* [B,C siblings; D,E siblings]
- *Root* [A]
- *Leaf* [D,E,C]
- *Ancestor* [A,B ancestors of D,E; A ancestor of B,C]
- *Descendant* [D,E descendants of A,B; B,C descendants of A]

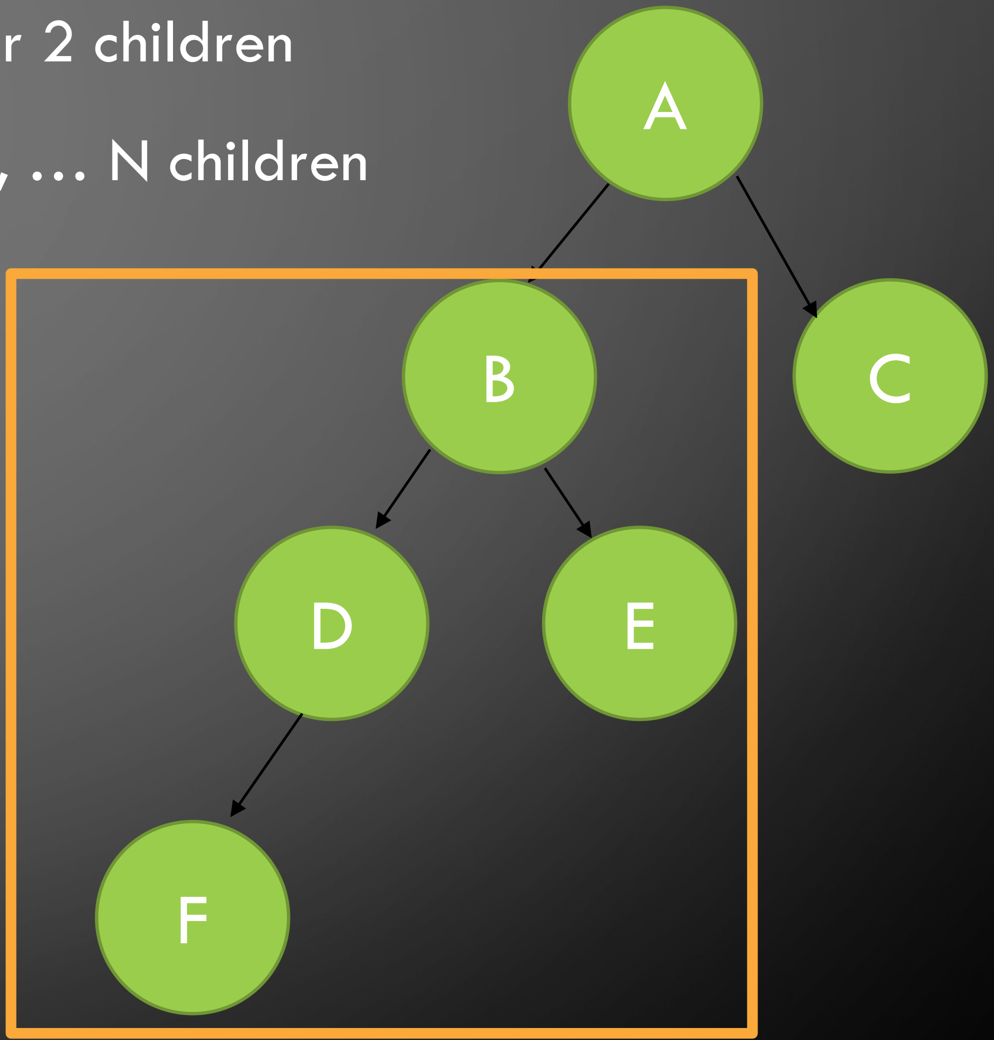


Terminology: Types of Trees

- Binary Tree: One root, each node has 0, 1, or 2 children
- N-ary Tree: One root, each node has 0, 1, 2, … N children
- Subtree: A node and its decedents



Ternary (3) Tree

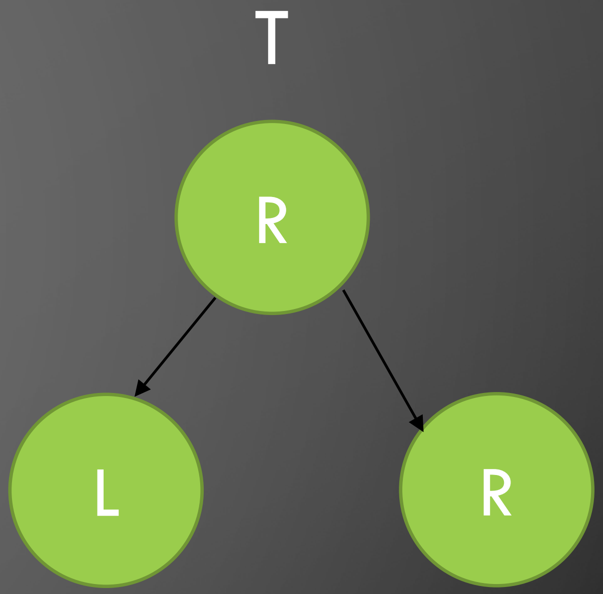


Binary Tree

Formal Definition of a Binary Tree

- **A Binary Tree T is a set of nodes such that**
- T is empty (or)
- T is partitioned into three subsets:
    1. A single node R, the root

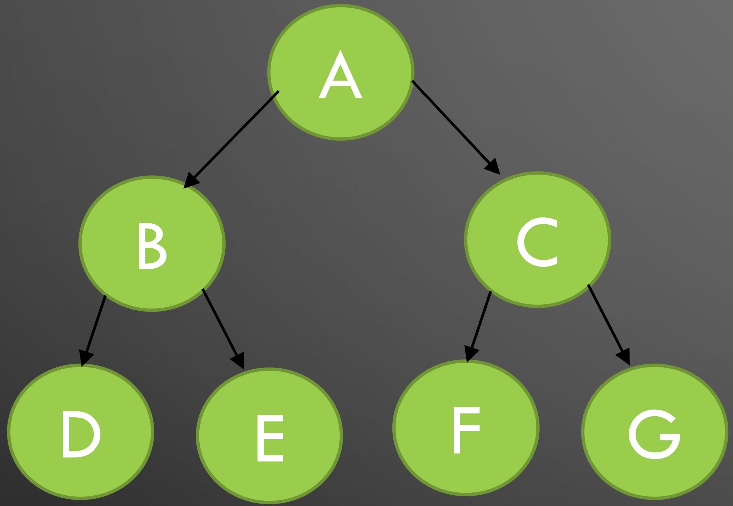
       Two (possibly empty) sets forming binary subtrees
    2. The left subtree
    3. The right subtree



Binary Tree Terminology

- Path
- Height: maximum length from root to a leaf

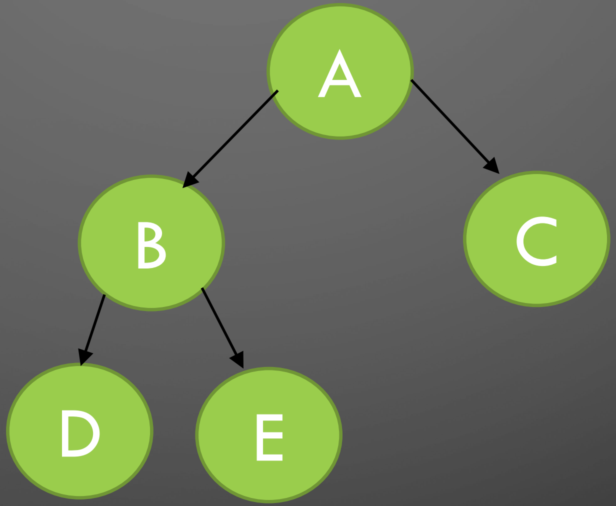
Full Tree



Every node has 2 children up until the height of the tree; all nodes at the height of the tree have 0 children.

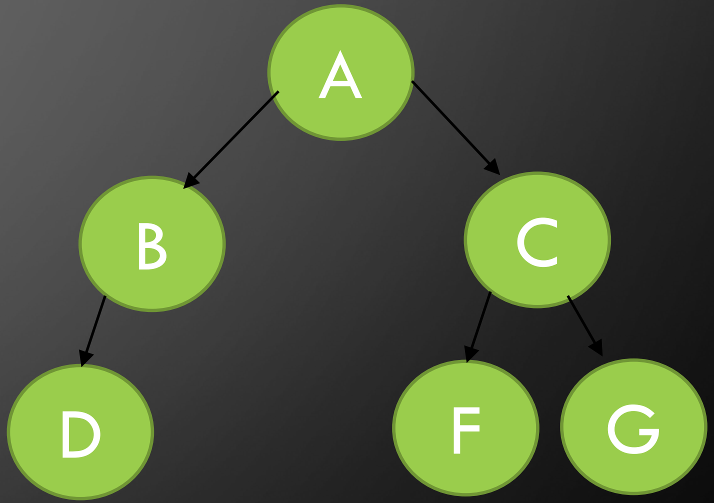
A full tree is both complete and balanced.

Complete Tree



The level above the height of the tree is full; all nodes are as far left as possible

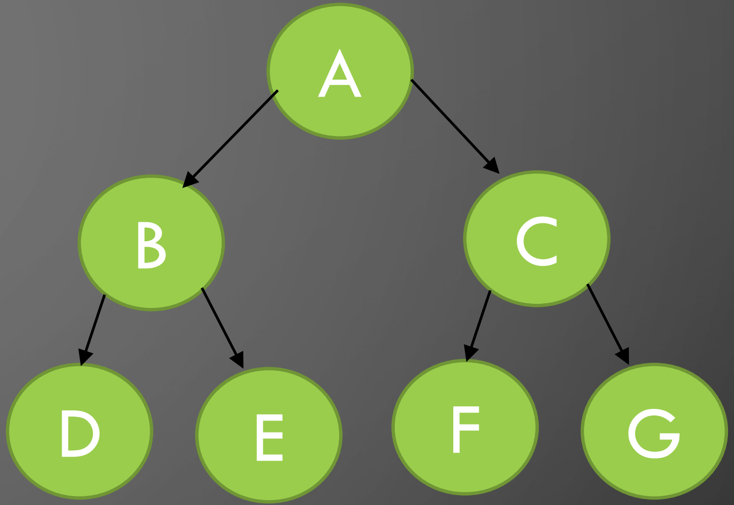
Balanced Tree



For each node, difference of heights of right and left subtree are within 1

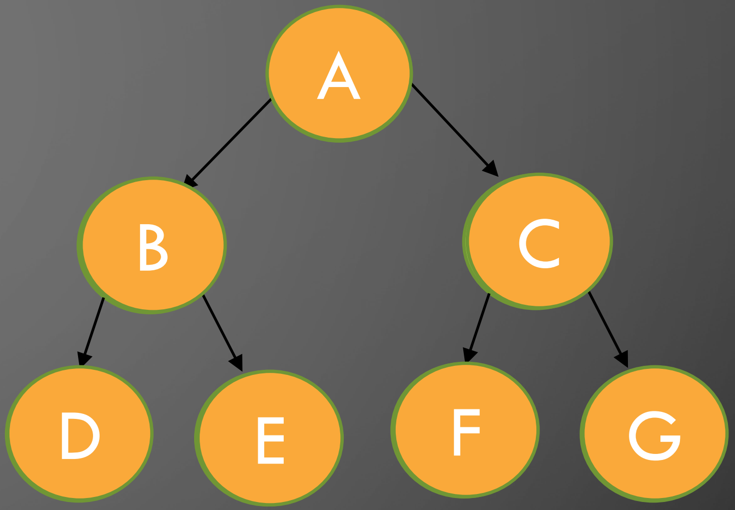
Operations on Trees

- attachRightSubtree()
  - Similarly attachLeftSubtree(), attachRight(), attachLeft()
- detachLeftSubtree()
  - Similarly detachRightSubtree()
- getLeftSubtree(), getRightSubTree()
- createTree(), destroyBinaryTree(), isEmpty()
- getRootData(), setRootData()
- PreorderTraverse(), inorderTraverse(), postorderTraverse()
  - Next slides



Traversals of Binary Trees

- *Preorder traversal*
  - If Tree is not empty
    - Visit the root of T
    - Preorder traverse left subtree of T
    - Preorder traverse right subtree of T



- What is the order these nodes are visited in?

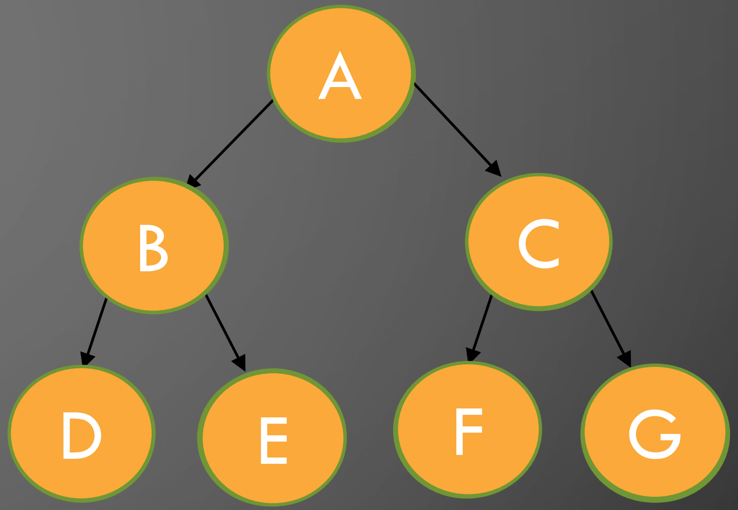
ORDER: A, B, D, E, C, F, G

🟢 Not traversed

🟠 Visited

Traversals of Binary Trees

- *In-order traversal*
  - If Tree is not empty
    - In-order traverse left subtree of T
    - Visit the root of T
    - In-order traverse right subtree of T



- What is the order these nodes are visited in?

ORDER: D, B, E, A, F, C, G,

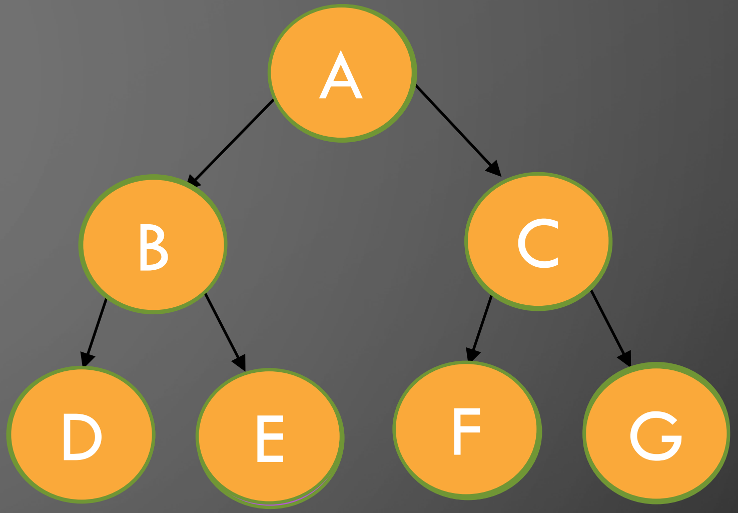
🟢 Not traversed

🔵 Traversed left subtree

🟠 Visited

Traversals of Binary Trees

- *Postorder traversal*
  - If Tree is not empty
    - postorder traverse left subtree of T
    - postorder traverse right subtree of T
    - Visit the root of T

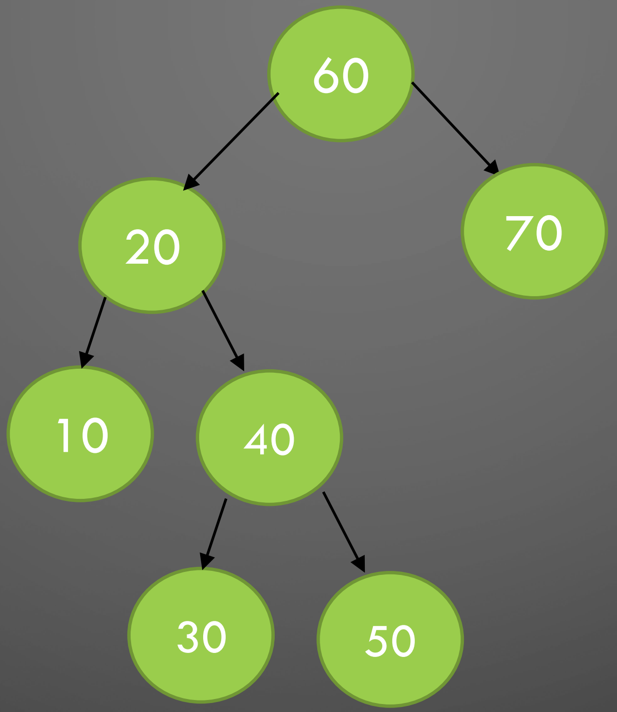


- What is the order these nodes are visited in?

ORDER: D, E, B, F, G, C, A,

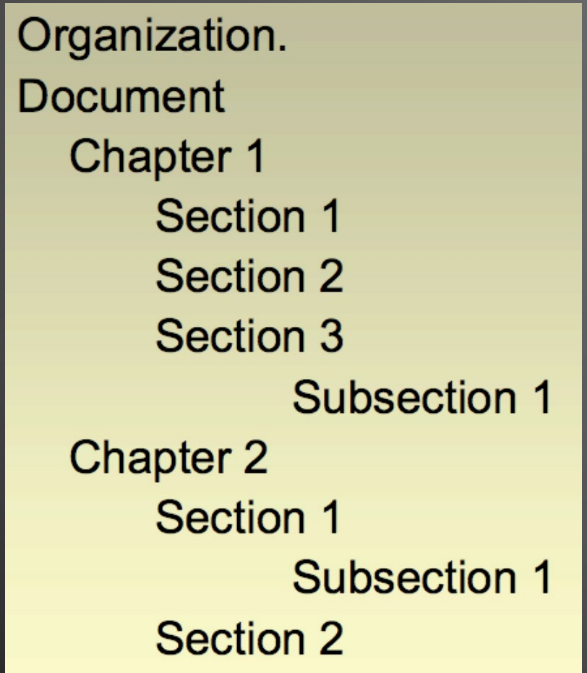
🟩 Not traversed 🟦 Traversed left subtree 🟪 Traversed right subtree 🟧 Visited

Example



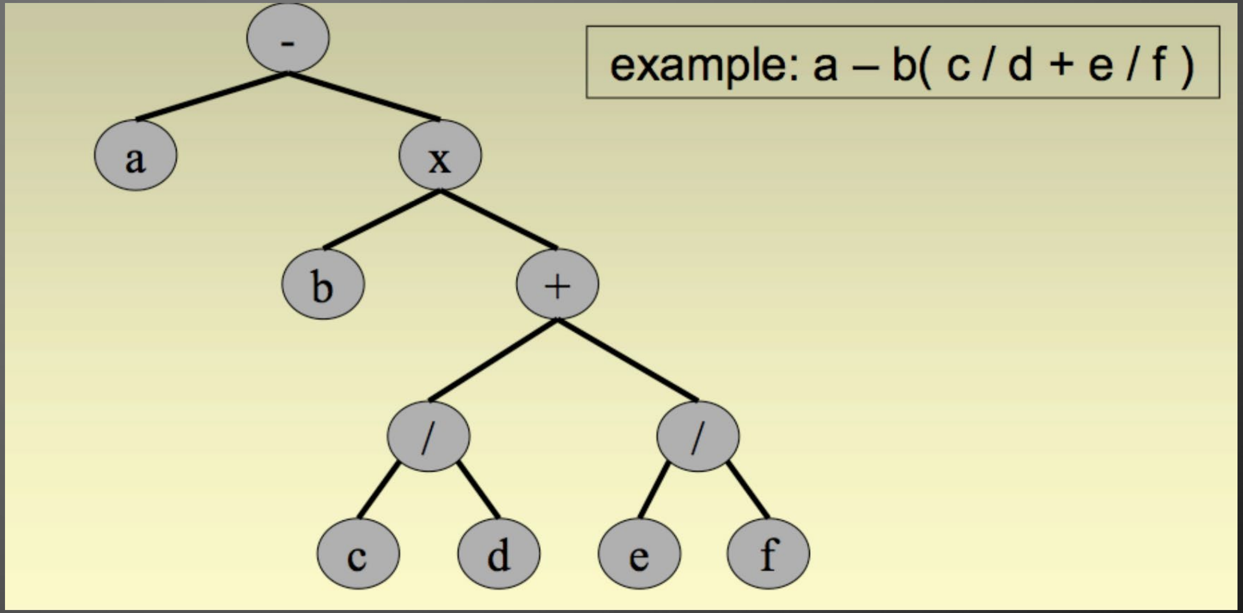
Preorder: 60 20 10 40 30 50 70 In order: 10 20 30 40 50 60 70 Postorder: 10 30 50 40 20 70 60

Usefulness of Trees



- Organization.
- Document
  - Chapter 1
    - Section 1
    - Section 2
    - Section 3
      - Subsection 1
  - Chapter 2
    - Section 1
      - Subsection 1
    - Section 2

Table of Contents

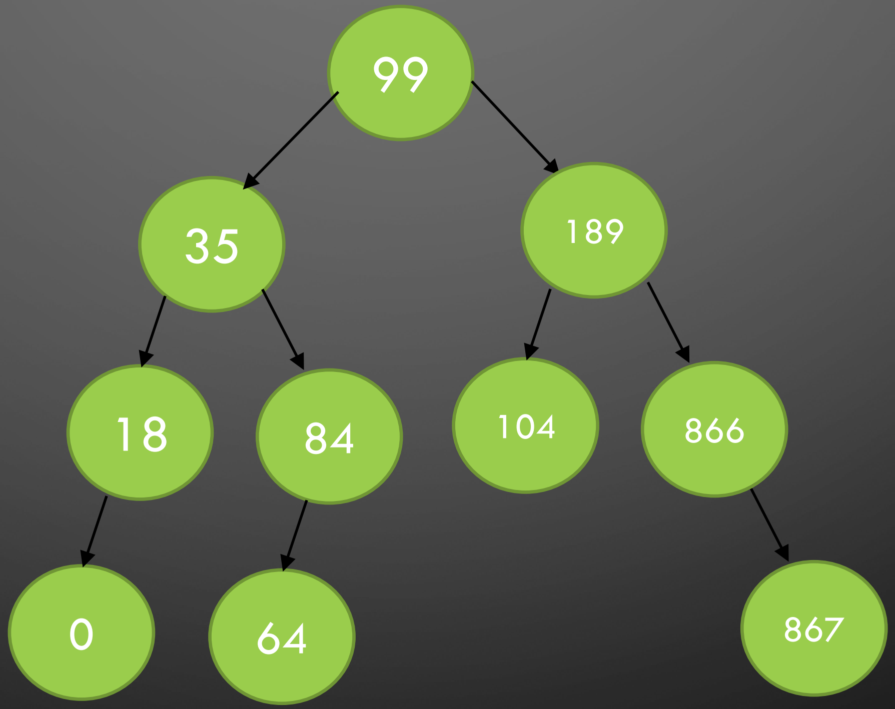


example: a – b(c / d + e / f)

Parsing

Usefulness of Trees: Organization and Searching

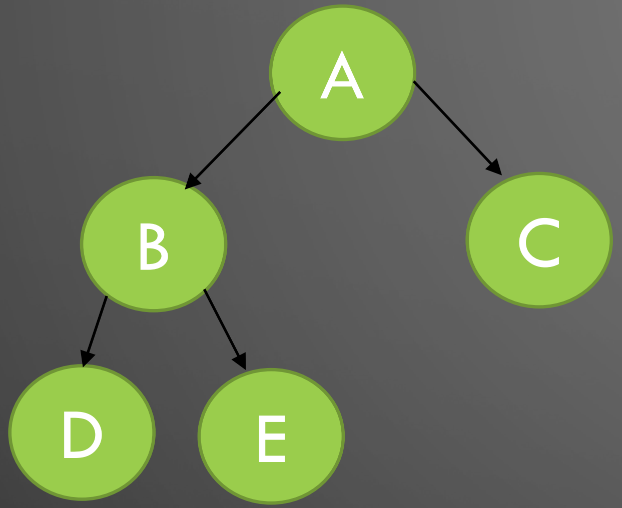
0	18	35	64	84	99	104	189	866	867



If n < root go left; if n > root go right

Representing Binary Trees

- Array-Based implementation only works for complete trees

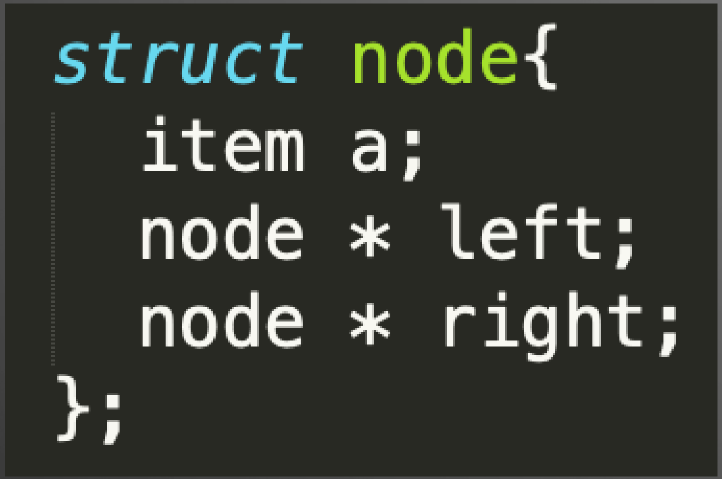


A	B	C	D	E

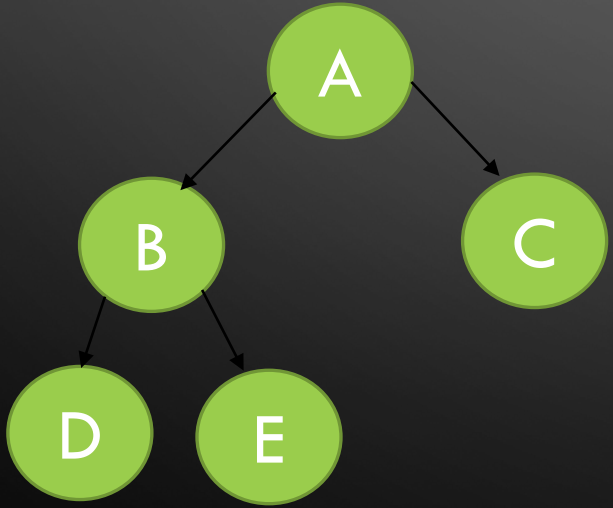
Completeness requirement:
So that the hierarchy is recoverable from the array

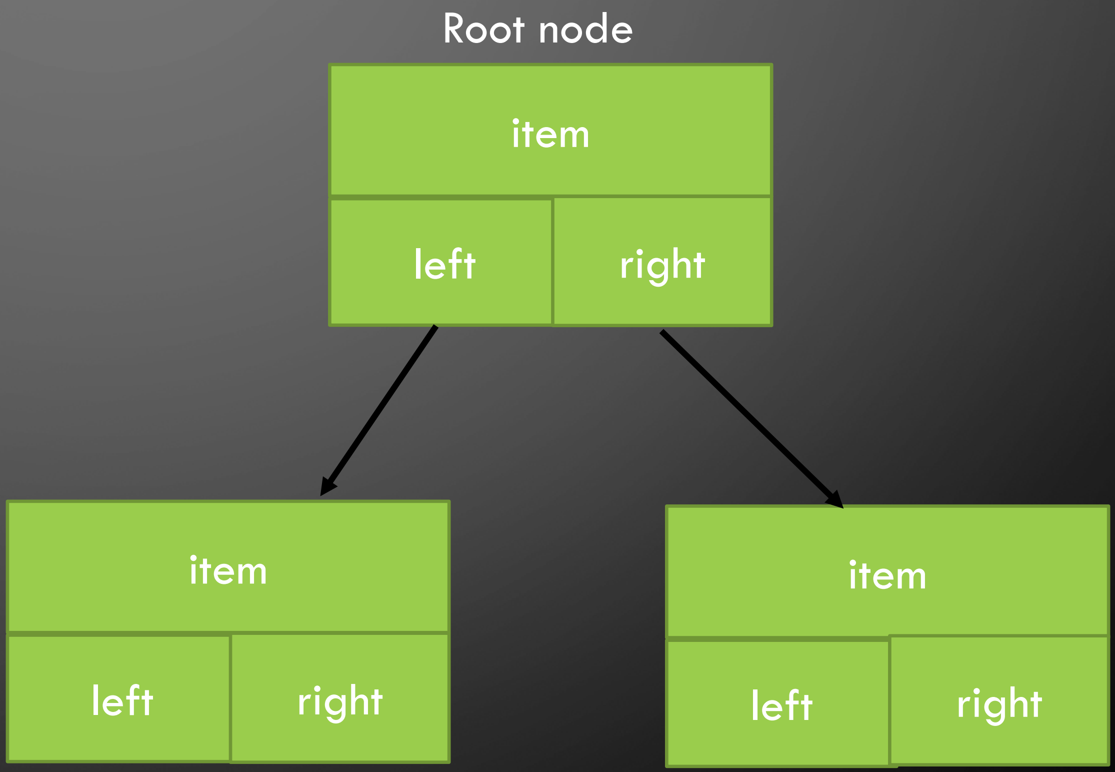
Representing Binary Trees

- Pointer-Based implementation- extends linked-list



```
struct node{
   item a;
   node * left;
   node * right;
};
```





Assignment/Homework

- Reading pp. 475-509
- ICE 7 Queue and Ring Buffer due on Tuesday.
- ICE 8 Priority Queue will be released today.