

CSE1303 Part A
Data Structures and Algorithms
Summer Semester 2003

Lecture A12 – Binary Trees

Kymerly Fergusson

Overview

- Trees.
- Terminology.
- Traversal of Binary Trees.
- Expression Trees.
- Binary Search Trees.

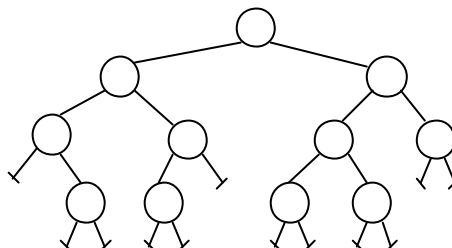
2

Trees

- Family Trees.
- Organisation Structure Charts.
- Program Design.
- Structure of a chapter in a book.

3

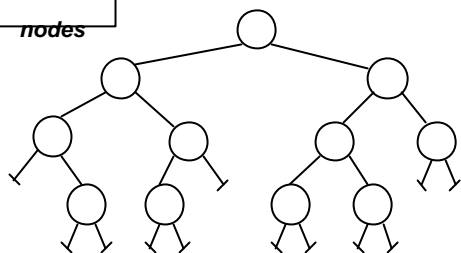
Parts of a Tree



4

Parts of a Tree

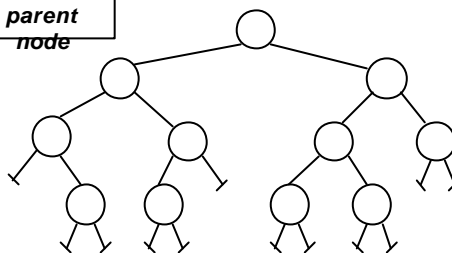
nodes



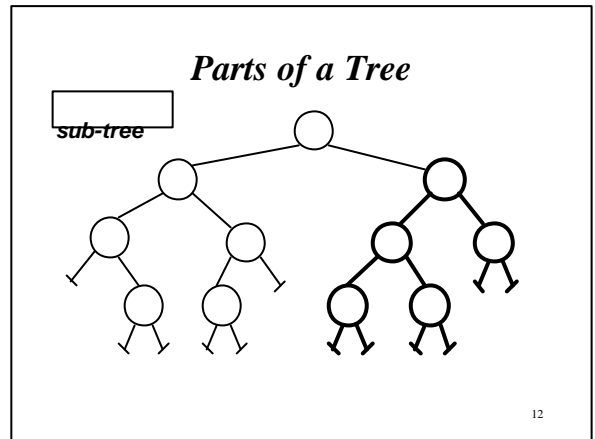
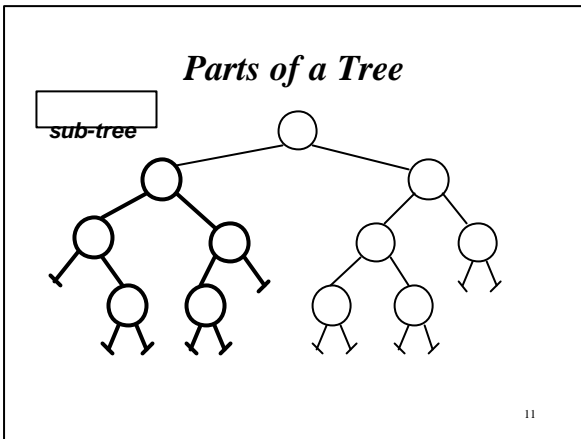
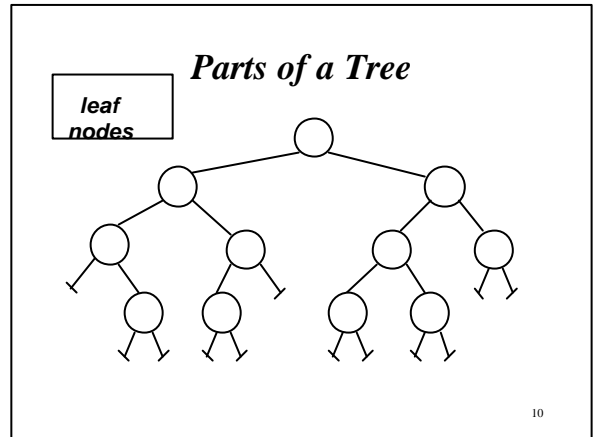
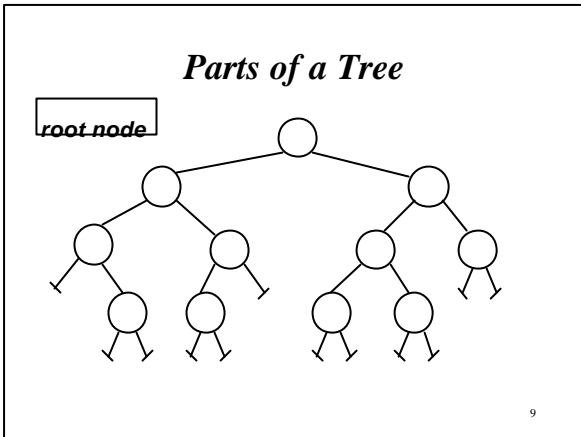
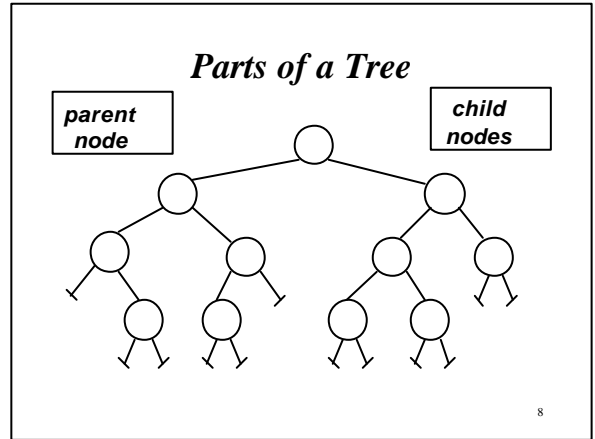
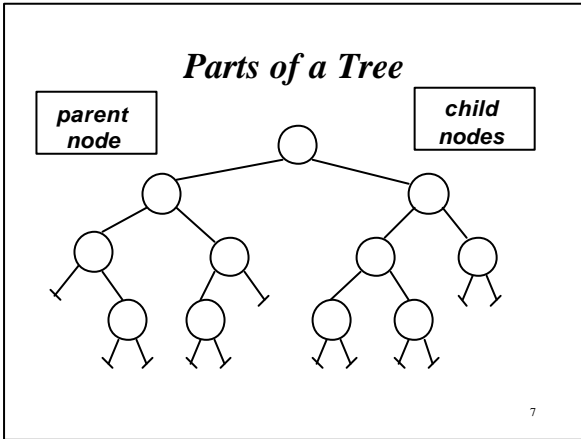
5

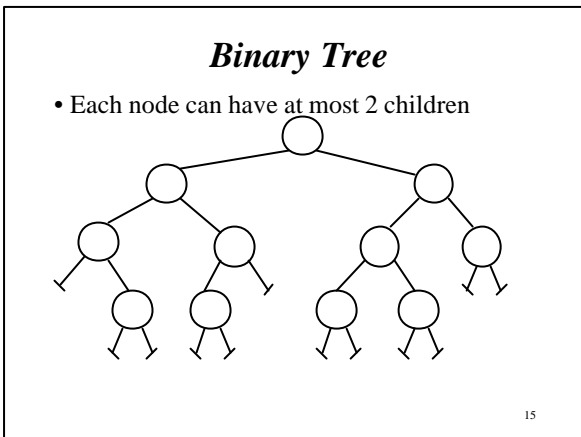
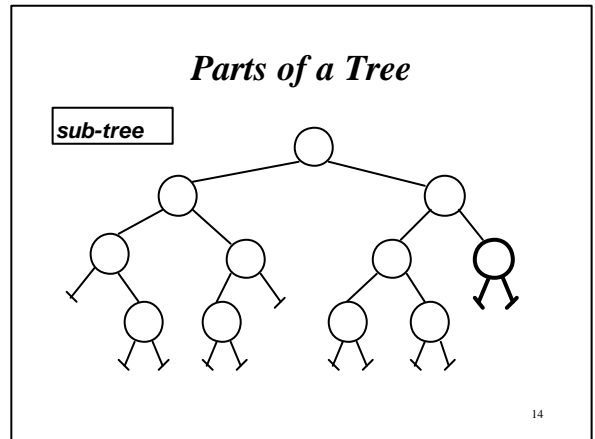
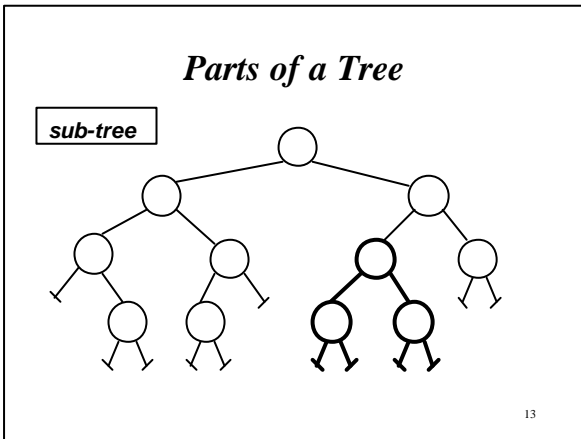
Parts of a Tree

parent node



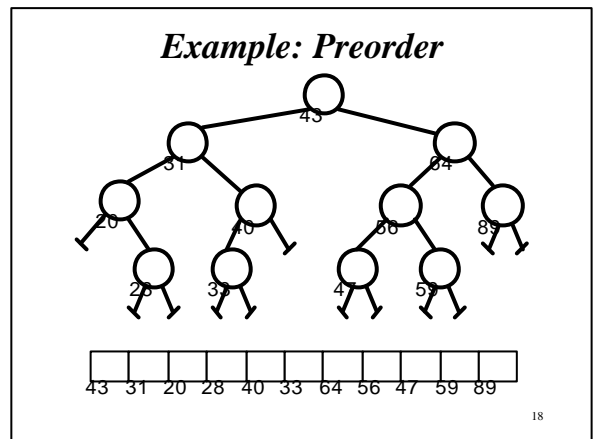
6





- ### Traversal
- Systematic way of visiting all the nodes.
 - Methods:
 - Preorder, Inorder, and Postorder
 - They all traverse the left subtree before the right subtree.
 - The name of the traversal method depends on when the node is visited.
- 16

- ### Preorder Traversal
- Visit the node.
 - Traverse the left subtree.
 - Traverse the right subtree.
- 17

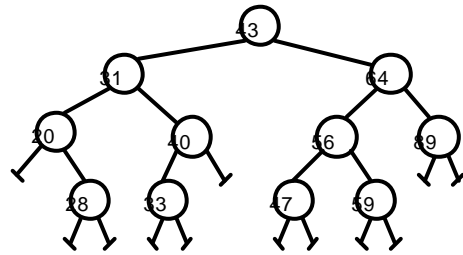


Inorder Traversal

- Traverse the left subtree.
- Visit the node.
- Traverse the right subtree.

19

Example: Inorder



20 28 31 33 40 43 47 56 59 64 89

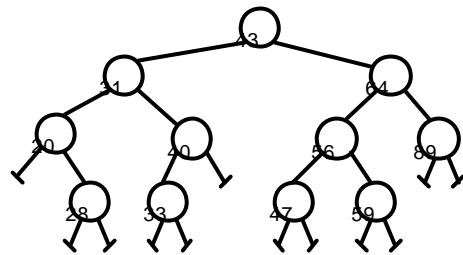
20

Postorder Traversal

- Traverse the left subtree.
- Traverse the right subtree.
- Visit the node.

21

Example: Postorder



28 20 33 40 31 47 59 56 89 64 43

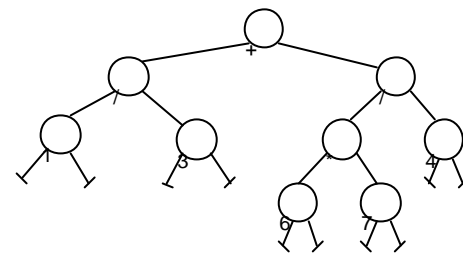
22

Expression Tree

- A Binary Tree built with operands and operators.
- Also known as a parse tree.
- Used in compilers.

23

Example: Expression Tree



1/3 + 6*7/4

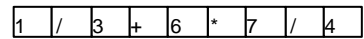
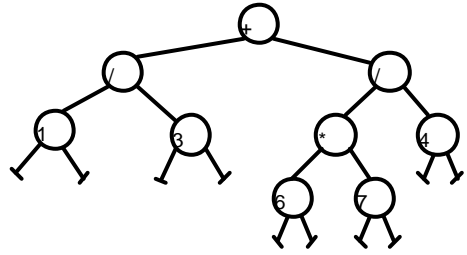
24

Notation

- Preorder
 - Prefix Notation
- Inorder
 - Infix Notation
- Postorder
 - Postfix Notation

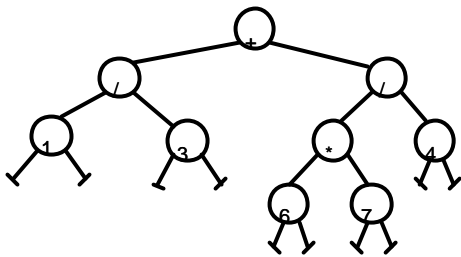
25

Example: Infix

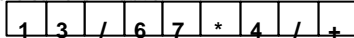


26

Example: Postfix

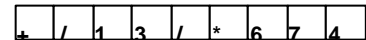
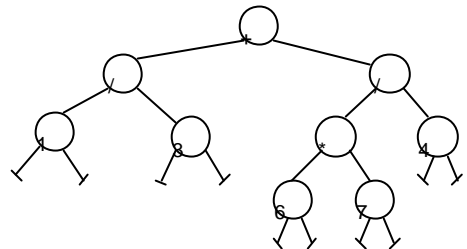


Recall: Reverse Polish Notation



27

Example: Prefix



28

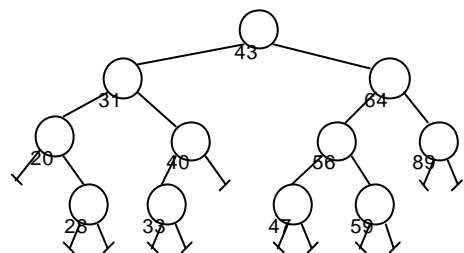
Binary Search Tree

A Binary Tree such that:

- Every node entry has a **unique** key.
- **All** the keys in the **left subtree** of a node are **less** than the key of the node.
- **All** the keys in the **right subtree** of a node are **greater** than the key of the node.

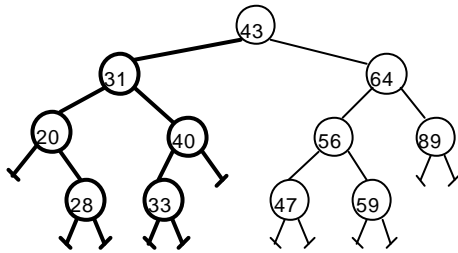
29

Example 1: key is an integer



30

Example 1: *key is an integer*



31

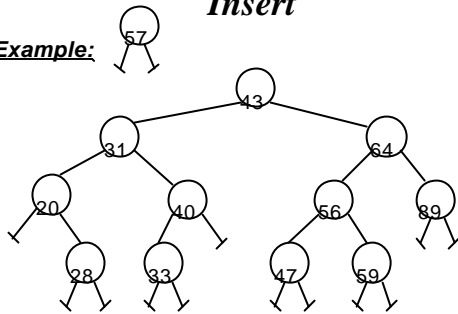
Insert

- Create new node for the item.
- Find a parent node.
- Attach new node as a leaf.

32

Insert

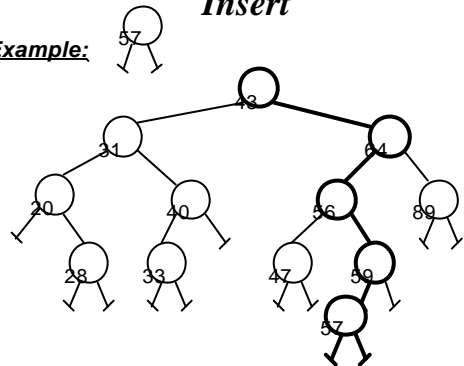
Example:



33

Insert

Example:



34

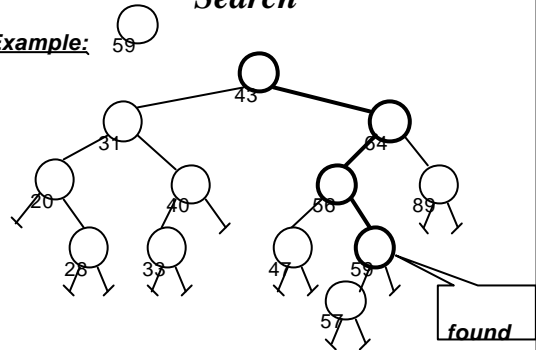
Search: Checklist

- if target key is less than current node's key, search the left sub-tree.
- else, if target key is greater than current node's key, search the right sub-tree.
- returns:
 - if found, pointer to node containing target key.
 - otherwise, NULL pointer.

35

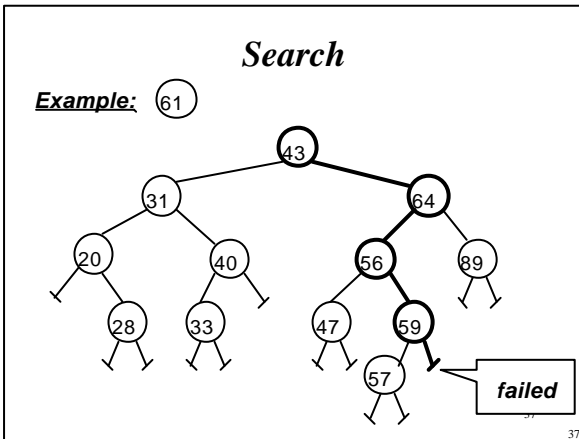
Search

Example:



36

36



- Revision**
- Binary Tree
 - Preorder, Inorder, Postorder Traversal
 - Expression Trees
 - Prefix, Infix, and Postfix notation
 - Insert and Search
- 38

- Revision: Reading**
- Kruse 9
 - Standish 9
 - Langsam 5
 - Deitel & Deitel 12.7
- Preparation**
- Next lecture: Binary Search Trees (Information Retrieval)*
- Read Chapter 9.2 in Kruse et al.
- 39