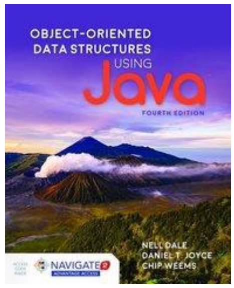
Chapter 7

The Binary Search Tree ADT



Section 2

BINARY TREES: TYPES AND TRAVERSAL ALGORITHMS

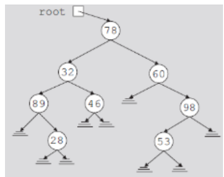
Objectives

- Learn about binary trees
- Study different types of binary trees including binary search trees
- Explore various binary search tree traversal algorithms

3

Binary Trees

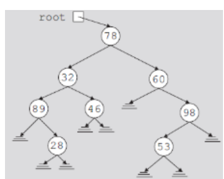
- A restricted class of trees: each inner node has no more than two children
- Nodes are represented as circles and are labeled with the data stored in them
- Directed edges



4

Binary Trees

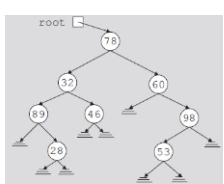
- Useful since they have a uniform structure
- The uniform structure allows efficient scanning and efficient access to data



5

Binary Trees Recursive Definition

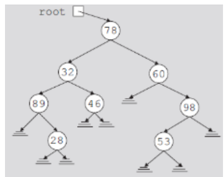
- A binary tree, T, is either empty or such that
  - T has a special node called the root node
  - T has two sets of nodes, L_T and R_T, called the left subtree and right subtree of T, respectively
  - L_T and R_T are binary trees



6

Left and Right Children

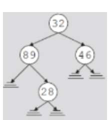
- Each node in a binary tree has 0, 1, or 2 children
- The node on the left is called the **left** child while the node on the right is referred to as the **right** child

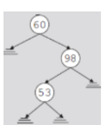


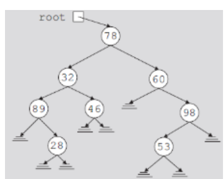
7

Left and Right Subtrees

- Each child node is the root of a subtree (**left subtree** and **right subtree**)



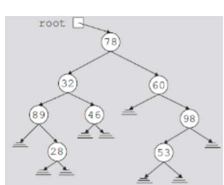




8

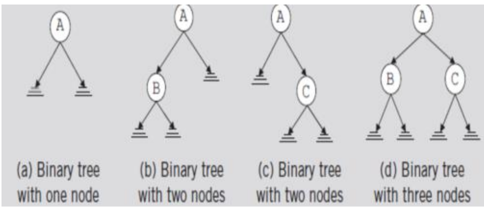
Drawing Binary Trees

1. Root node is drawn at the top
2. Left child of the root node (if any) is drawn below and to the left of the root node
3. Right child of the root node (if any) is drawn below and to the right of the root node



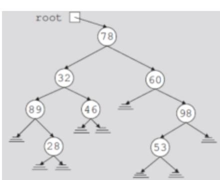
9

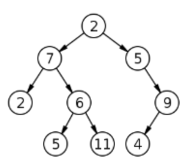
Binary Trees Example



10

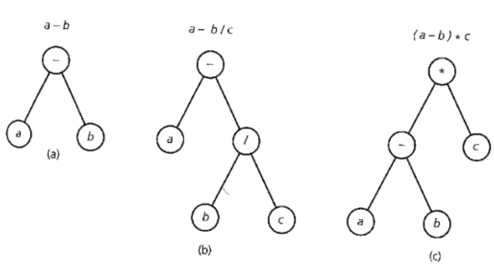
Binary Trees Example





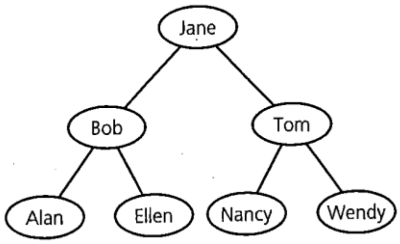
11

Binary Trees Example



12

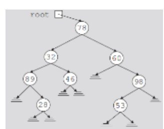
Binary Trees Example

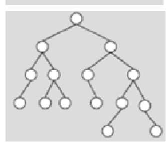


13

Density of Binary Tree

- At any level L, a binary tree contains from 1 to 2^L nodes
- The **density** of a tree is a measure of the size of a tree (number of nodes) relative to the height of the tree

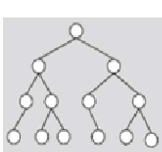


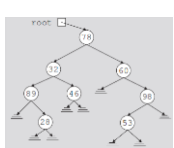


14

Dense Binary Trees

- Dense trees are good because they allow the storage of a large collection of data and maintain efficient access to the data (due to the shorter paths)

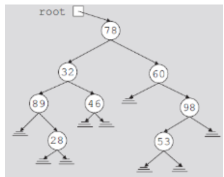




15

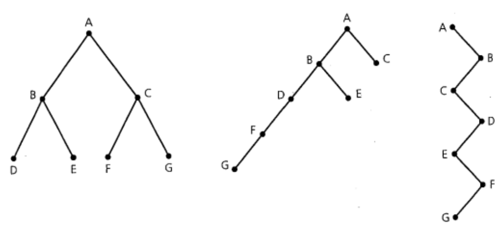
Binary Tree Minimum and Maximum Height

- The maximum height in a binary tree with N nodes is $H = N - 1$
- The minimum height in a binary tree with N nodes is $H = \lfloor \log_2 N \rfloor$



16

Binary Trees
(With Same Nodes But Different Heights)



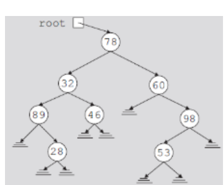
17

Number of Nodes

- The number of nodes N in a binary tree satisfies

$$2^H \leq N < 2^{H+1}$$

where H is the height of the tree



18

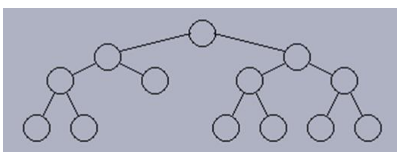
Types of Binary Trees

- Full Binary Trees
- Perfect Binary Trees
- Complete Binary Trees
- Binary Search Trees

19

Full Binary Trees

- A **full binary tree** is a binary tree in which every internal (non-leaf) node has two children

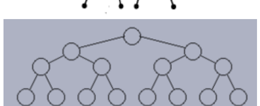


A Full Binary Tree

20

Perfect Binary Trees

- A **perfect binary tree** is a full binary tree in which all leaf nodes are at the same height or same level, and in which every parent has two children



A Perfect Binary Tree

21

Perfect Binary Trees : Number of Nodes

- A perfect binary tree contains 2^L nodes at each level L
- Thus, the number of leaf nodes (LN) in a perfect binary tree with height H is:

  $$LN = 2^H$$

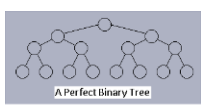
22

Perfect Binary Trees: Number of Nodes

- The number of nodes N in a perfect binary tree is:

  $$N = 2^{H+1} - 1$$

  where H is the height of the tree



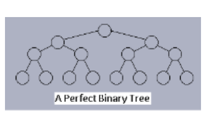
23

Perfect Binary Trees : Number of Nodes

- The number of nodes N in a perfect binary tree can also be found using the formula:

  $$N = 2\,LN - 1$$

  where LN is the number of leaf nodes in the tree



24

Complete Binary Trees

- A **complete binary tree** of height H is a binary tree in which each level 0 to $H-1$ has a full set of nodes and all leaf nodes at level H occupy the leftmost positions in the tree

25

Complete Binary Trees: Number of NULL Links

- The number of null links (absent children of nodes) in a complete binary tree of N nodes is:
  $N+1$

26

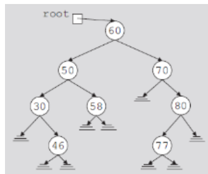
Complete Binary Trees: Number of Internal Nodes

- The number of internal (non-leaf) nodes in a complete binary tree of N nodes is:
  $\lfloor N/2 \rfloor$

27

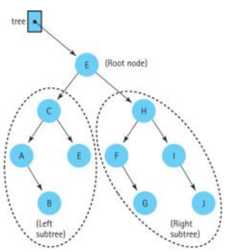
Binary Search Trees

- Data in each node of a binary search tree is
  - Larger than (or equal to, if duplication is allowed) the data in its left child
  - Smaller than the data in its right child



28

Binary Search Trees

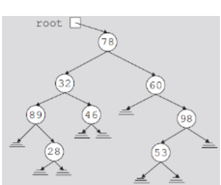


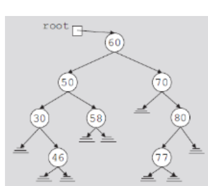
All values in the left subtree are less than or equal to the value in the root node.

All values in the right subtree are greater than the value in the root node.

29

Binary Search Trees vs. General Binary Trees





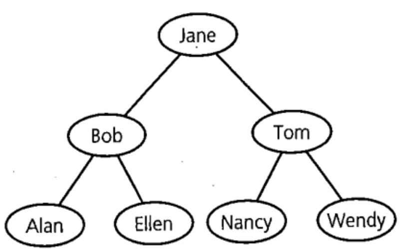
30

Binary Search Trees Recursive Definition

- A binary search tree, *T*, is either empty or the following is true:
  - *T* has a special node called the root node
  - *T* has two sets of nodes, L_T and R_T, called the left subtree and right subtree of T, respectively
  - The key in the root node is larger than every key in the left subtree and smaller than every key in the right subtree
  - L_T and R_T are binary search trees

31

Binary Search Trees: Example



32

Binary Search Tree Traversal

- The process of visiting (examining) each node in a binary tree exactly once and in a systematic way
- Binary search tree traversal is classified by the order in which the nodes are visited

33

Binary Search Tree Traversal

- Binary search tree traversal must start with the root, and then
  - Visit the node first *or*
  - Visit the subtrees first
- Three different traversal algorithms:
  1. Inorder
  2. Preorder
  3. Postorder

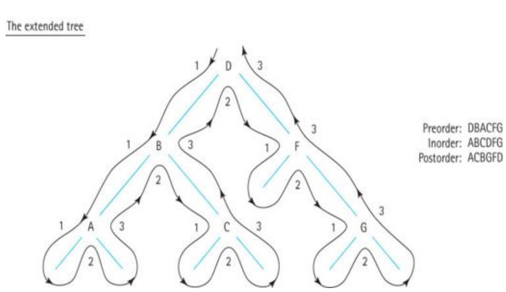
34

Traversals Definitions

- Preorder traversal
  - Visit the root, visit the left subtree, visit the right subtree
- Inorder traversal
  - Visit the left subtree, visit the root, visit the right subtree
- Postorder traversal
  - Visit the left subtree, visit the right subtree, visit the root

35

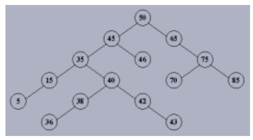
Visualizing Traversals



36

Inorder Tree Traversal

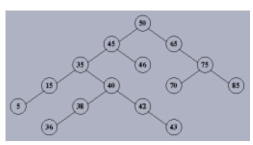
- (LNR) Traversal
  - Traverse the left subtree
  - Visit the node
  - Traverse the right subtree
- Example:
  - 5 15 35 36 38 40 42 43 45 46 50 65 70 75 85



37

Preorder Tree Traversal

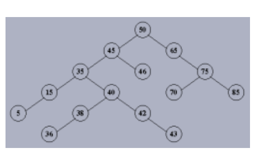
- (NLR) Traversal
  - Visit the node
  - Traverse the left subtree
  - Traverse the right subtree
- Example:
  - 50 45 35 15 5 40 38 36 42 43 46 65 75 70 85



38

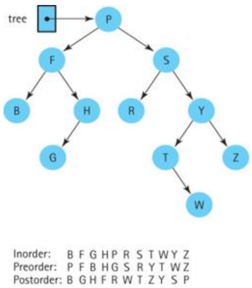
Postorder Tree Traversal

- (LRN) Traversal
  - Traverse the left subtree
  - Traverse the right subtree
  - Visit the node
- Example:
  - 5 15 36 38 43 42 40 35 46 45 70 85 75 65 50



39

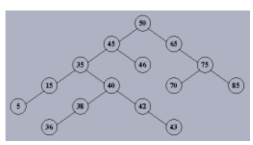
Three Binary Search Tree Traversals



40

Breadth First Scan

- Scan the tree level by level starting at the root and moving to the first generation of children, then the second generation, and so forth
- A queue is used to store the nodes as they are being visited

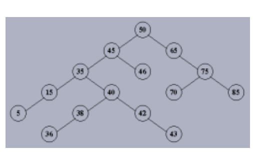


41

Breadth First Scan

- Example:

50 45 65 35 46 75
15 40 70 85 5
38 42 36 43



42