DATA STRUCTURES AND ITS APPLICATIONS

BST Implementation using Dynamic Allocation : Insertion

Shylaja S S & Kusuma K V

Department of Computer Science
& Engineering

DATA STRUCTURES AND ITS APPLICATIONS

BST Implementation using Dynamic Allocation: Insertion

Shylaja S S

Department of Computer Science & Engineering

Binary Search Tree – An Application of Binary Tree

Background

Problem : find a target key in a list of elements

Sequential: Potentially enumerate every key

Ordered List: Searching can be done on logn

Frequent insertions and deletions : Ordered List is much slower

Solution: Binary Trees provide an excellent solution to this by organizing every element in the list as a node in the tree

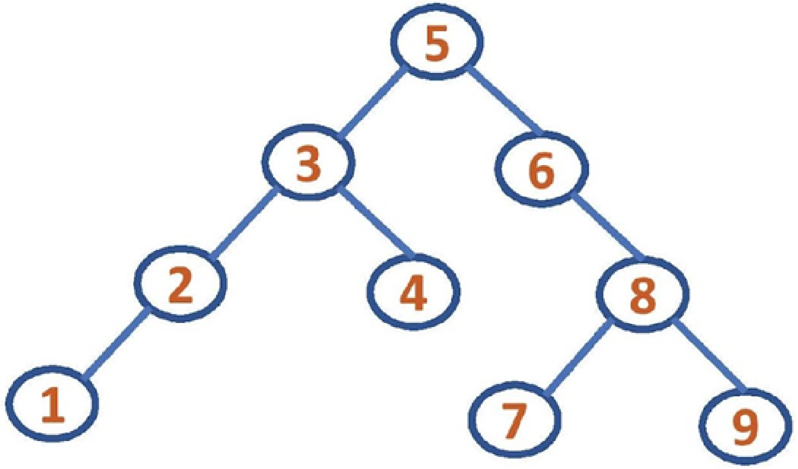
Binary Search Tree: Definition

A Binary Search Tree is a binary tree which has the following properties:

- all the elements in the left subtree of a node **n** are less than the contents of node **n**
- all the elements in the right subtree of a node **n** are greater than or equal to the contents of node **n**

Binary Search Tree – An Application of Binary Tree

A Binary Search Tree with the nodes inserted in the order: 5, 3, 6, 4, 2, 8, 1,7, 9



Binary Search Tree - Implementation

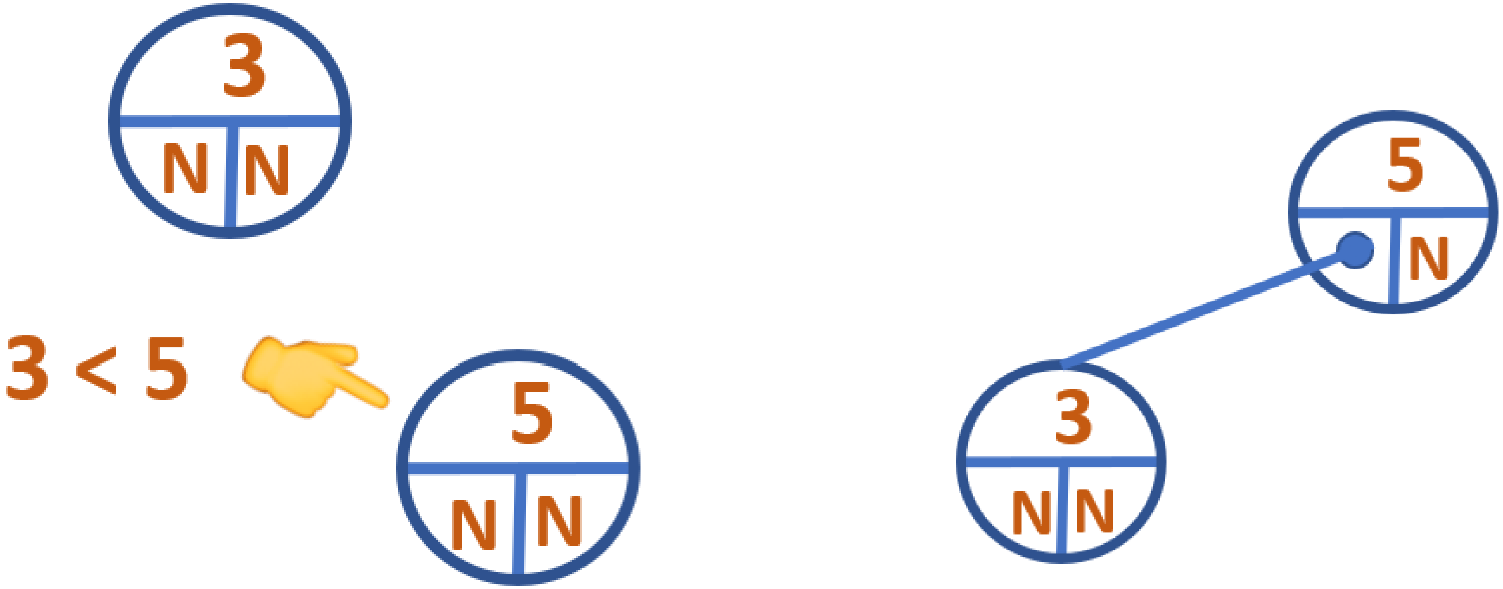
Linked implementation

Here every node will have its own **info** along with the **links to left child** and **right child**

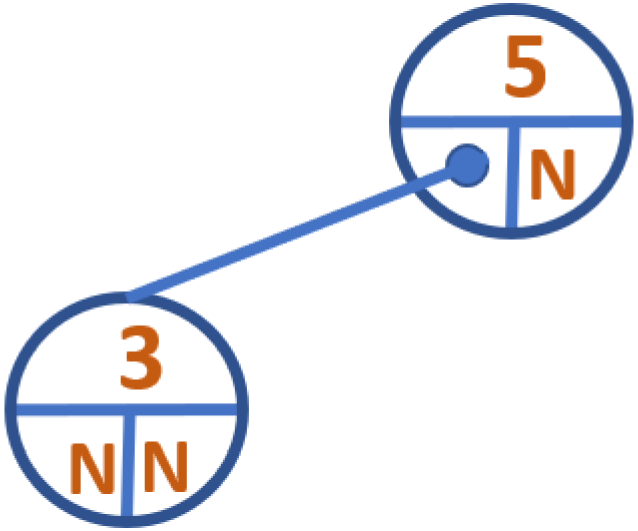
```
typedef struct tree_linked
{
 int info;
 struct tree_linked *left,*right;
}NODE;

NODE *root=NULL; //root points to Root of the tree and
initially it is null
```

Linked implementation: 5, 3, 7, 8, 1, 4

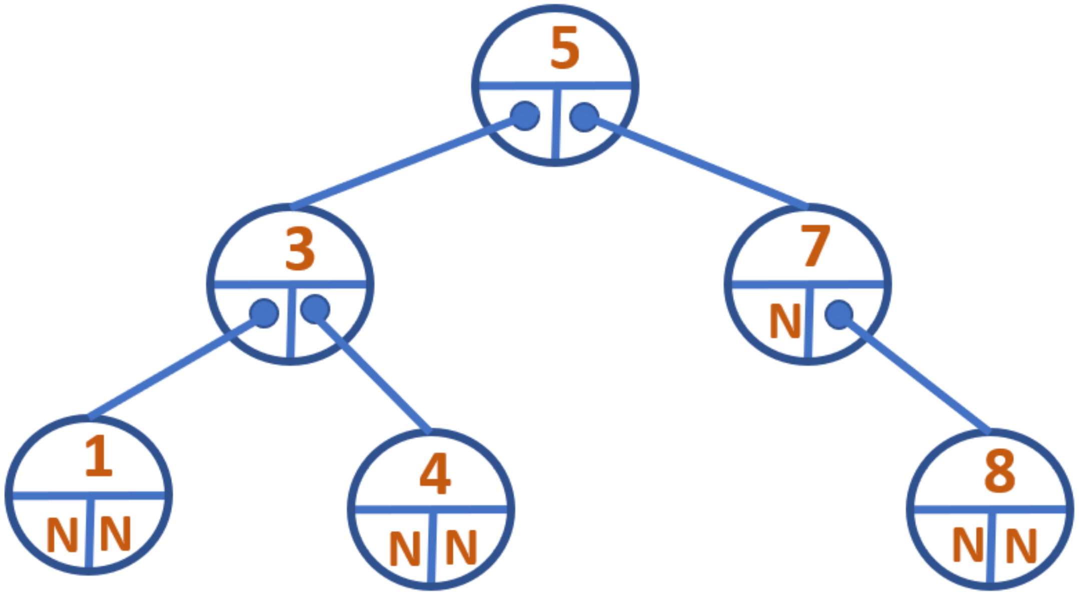


Linked implementation: 5, 3, 7, 8, 1, 4



Binary Search Tree - Implementation

Linked implementation: 5, 3, 7, 8, 1, 4



1. Which of the following is the defining property of a Binary Search Tree (BST)?

A) All left child nodes have keys greater than their parent.

B) All right child nodes have keys smaller than their parent.

C) The left subtree of a node contains only nodes with keys less than the node's key, and the right subtree contains only nodes with keys greater.

D) Nodes are always stored in level-order.

1. Which of the following is the defining property of a Binary Search Tree (BST)?

A) All left child nodes have keys greater than their parent.

B) All right child nodes have keys smaller than their parent.

C) The left subtree of a node contains only nodes with keys less than the node’s key, and the right subtree contains only nodes with keys greater.

D) Nodes are always stored in level-order.

2. When inserting a new key into a BST:

A) It always becomes a leaf node.

B) It always replaces the root if smaller.

C) It is inserted at the leftmost or rightmost node directly.

D) It must be inserted only in the left subtree.

2. When inserting a new key into a BST:

A) It always becomes a leaf node.

B) It always replaces the root if smaller.

C) It is inserted at the leftmost or rightmost node directly.

D) It must be inserted only in the left subtree.

3. If the node to be deleted in a BST has two children, how is deletion handled?

A) Node is simply removed, and its children are discarded.

B) Replace the node with its inorder predecessor or inorder successor.

C) Replace the node with the root node.

D) Deletion is not possible for such a node.

3. If the node to be deleted in a BST has two children, how is deletion handled?

A) Node is simply removed, and its children are discarded.

B) Replace the node with its inorder predecessor or inorder successor.

C) Replace the node with the root node.

D) Deletion is not possible for such a node.

4. Which of the following is NOT a typical application of BSTs?

A) Searching

B) Sorting

C) Priority Queue implementation

D) Symbol Table in Compilers

4. Which of the following is NOT a typical application of BSTs?

A) Searching

B) Sorting

C) Priority Queue implementation

D) Symbol Table in Compilers

5. Which of the following is a balanced version of a BST?

A) AVL Tree

B) Red-Black Tree

C) B-Tree

D) All of the above

5. Which of the following is a balanced version of a BST?

A) AVL Tree

B) Red-Black Tree

C) B-Tree

D) All of the above

PES
UNIVERSITY

THANK YOU

Shylaja S S

Department of Computer Science
& Engineering

shylaja.sharath@pes.edu

+91 9449867804