SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IOT Including CS&BCT

COURSE NAME : 19CS307- DATA STRUCTURES

II YEAR / III SEMESTER

Unit III- **NON LINEAR DATA STRUCTURES - Tree**

Topic 6 : Binary Search Tree ADT

Problem

- Draw the binary search tree for the following input : 14,15,4,9,7,18,3,5,16,4,20,17,9,14,5

Binary Search Tree

- Binary search tree is a data structure that quickly allows us to maintain a sorted list of numbers.
- It is called a binary tree because each tree node has maximum of two children.
- (OR)
- A binary search tree (BST), also known as an ordered binary tree, is a node-based data structure in which each node has no more than two child nodes.
- Each child must either be a leaf node or the root of another binary search tree.
- It is called a search tree because it can be used to search for the presence of a number in O(log(n)) time.
- The properties that separates a binary search tree from a regular binary tree is
- All nodes of left subtree are less than root node
- All nodes of right subtree are more than root node
- Both subtree of each node are also BSTs i.e. they have the above two properties

Binary Search Tree –Cont..

Basic Operations

Following are the basic operations of a tree –

Search – Searches an element in a tree.

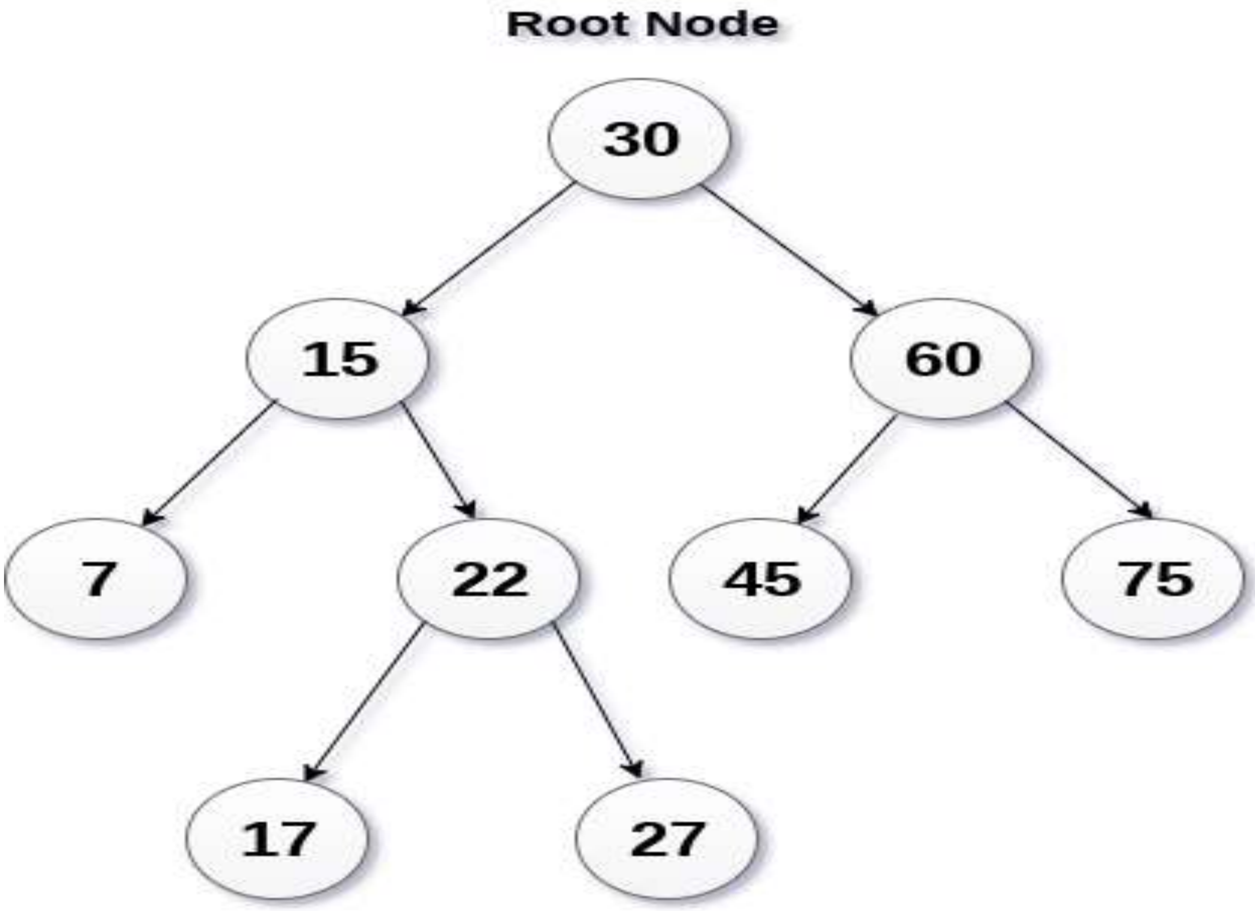
Insert – Inserts an element in a tree.

Pre-order Traversal – Traverses a tree in a pre-order manner.

In-order Traversal – Traverses a tree in an in-order manner.

Post-order Traversal – Traverses a tree in a post-order manner.

Binary Search Tree –Cont..

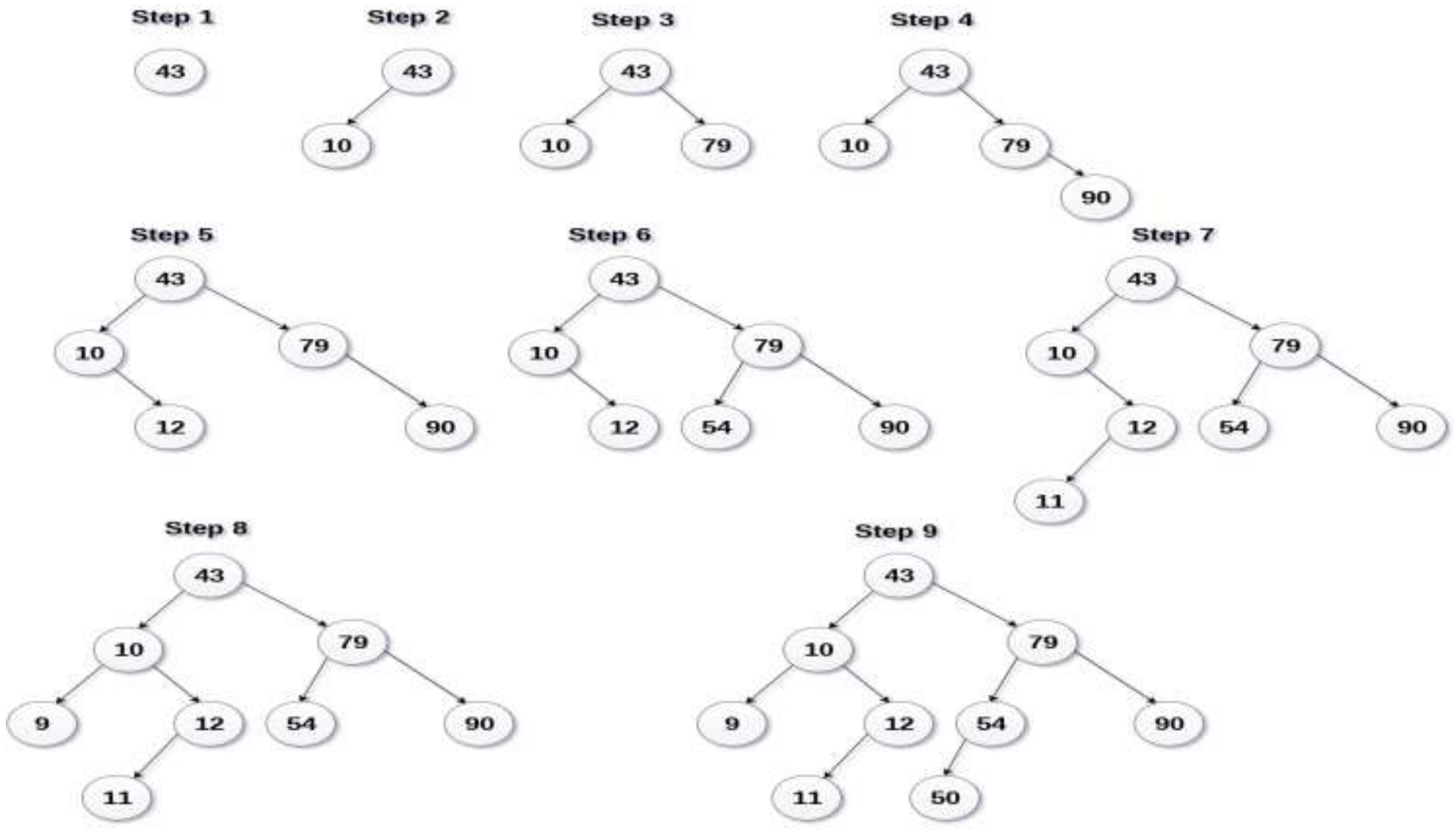


Binary Search Tree

Binary Search Tree –Cont..

- Create the binary search tree using the following data elements.
- **43, 10, 79, 90, 12, 54, 11, 9, 50**
- Insert 43 into the tree as the root of the tree.
- Read the next element, if it is lesser than the root node element, insert it as the root of the left sub-tree.
- Otherwise, insert it as the root of the right of the right sub-tree.
- The process of creating BST by using the given elements, is shown in the image below.

Binary Search Tree –Cont..



Binary search Tree Creation

Binary Search Tree –Cont..

Node

Define a node having some data, references to its left and right child nodes.

```
struct node
{
int data;
struct node *leftChild;
struct node *rightChild;
};
```

Binary Search Tree –Cont..

- Search Operation
- Algorithm
- struct node* search(int data)
- {
- struct node *current = root;
- printf("Visiting elements: ");
- while(current->data != data)
- {
- if(current != NULL)
- {

```
Printf("%d ",current->data);
//go to left tree
if(current->data > data)
{
 current = current->leftChild;
}
//else go to right tree else
{
 current = current->rightChild;
}
//not found
if(current == NULL)
{  return NULL; }
 } }return current; }
```

Binary Search Tree –Cont..
Insert Operation

```
void insert(int data)
{
struct node *tempNode = (struct node*) malloc(sizeof(struct node));
struct node *current;
struct node *parent;
tempNode->data = data;
tempNode->leftChild = NULL;
tempNode->rightChild = NULL;
```

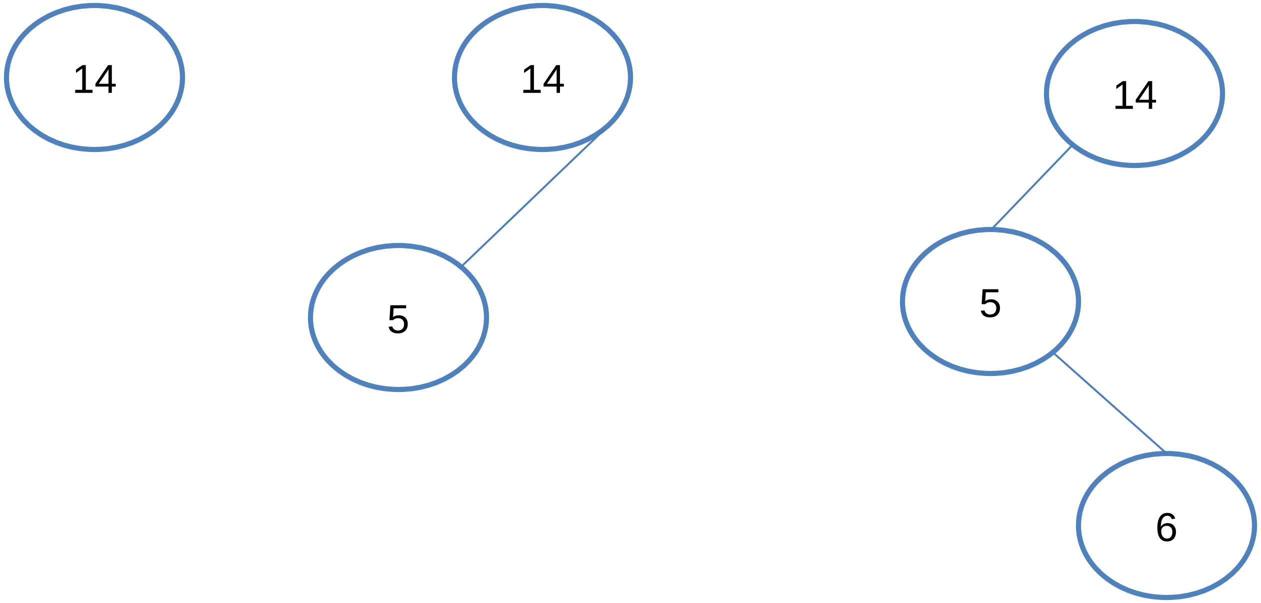
```
//if tree is empty
if(root == NULL)
{
root = tempNode;
}
else
{
current = root; parent = NULL;
while(1)
{
parent = current;
```

Binary Search Tree –Cont..

- //go to left of the tree
- if(data < parent->data)
- {
- current = current->leftChild;
- //insert to the left
- if(current == NULL)
- {
- parent->leftChild = tempNode;
- return;
- }
- }
- //go to right of the tree else
- {
- current = current->rightChild;
- //insert to the right
- if(current == NULL)
- {
- parent->rightChild = tempNode;
- return;
- } } } } }

Binary Search Tree –Cont..

Draw the binary search tree for the following input : 14,5,6,2,18,20,16,18,-1,21



Activity

MCQ

1. Which of the following traversal outputs the data in sorted order in a BST?

(A) Pre order
(B) In order
(C) Post order
(D) Level order

2. What does the following piece of code do?

```
public void func(Tree root)
  {
   System.out.println(root.data());
   Func(root.left());
   func(root.right());
   }
```

a) Preorder traversal
b) b) Inorder traversal
c) c) Postorder traversal
d) d) Level order traversal

Advantages

- ➢Searching become very efficient in a binary search tree since, we get a hint at each step, about which sub-tree contains the desired element.
- ➢The binary search tree is considered as efficient data structure in compare to arrays and linked lists. In searching process, it removes half sub-tree at every step. Searching for an element in a binary search tree takes o($\log_2 n$) time. In worst case, the time it takes to search an element is 0(n).
- ➢It also speed up the insertion and deletion operations as compare to that in array and linked list.

Disadvantages

- ➢Shape of the tree depends upon order of insertion and it can be degenerated.
- ➢Searching takes long time.

Assessment 1

1. List out the advantages of binary search tree

   a)____

   b)____

   c)____

   d)____

2. Identify the disadvantages of binary search tree

   a)____

   b)____

   c)____

   d)____

REFERENCES

1. M. A. Weiss, “Data Structures and Algorithm Analysis in C”, Pearson Education, 8th Edition, 2007. [Unit I, II, III, IV,V]

2. A. V. Aho, J. E. Hopcroft and J. D. Ullman, “Data Structures and Algorithms”, Pearson Education, 2nd Edition, 2007 [Unit IV].

3. A.M.Tenenbaum, Y. Langsam and M. J. Augenstein, “Data Structures using C”,PearsonEducation, 1st Edition, 2003.(UNIT I,II,V)

THANK YOU