DSA

23221

Page No. 1
Date.

Assignment No. 05

Title : Assignment 5 Binary Search tree

Aim : Implement a Binary search tree

problem statement : Implement binary search tree and perform following operations.

1) Insert (Handle duplicate entry)
2) Delete
3) Search
4) Display tree (Traversal)
5) Display - Depth of tree
6) Display - Mirror image
7) Create copy
8) Display all parent nodes with their child nodes
9) Display leaf nodes
10) Display tree level wise

Objective :

1) To study Data structure and their implementations and applications.
2) To learn different searching & sorting tech
3) to study some advanced data structures such as trees, graph and tables
4) To learn different file organizations
5) To learn algo development & analysis of algorithm

Scanned with CamScanner

Page No.
Date.

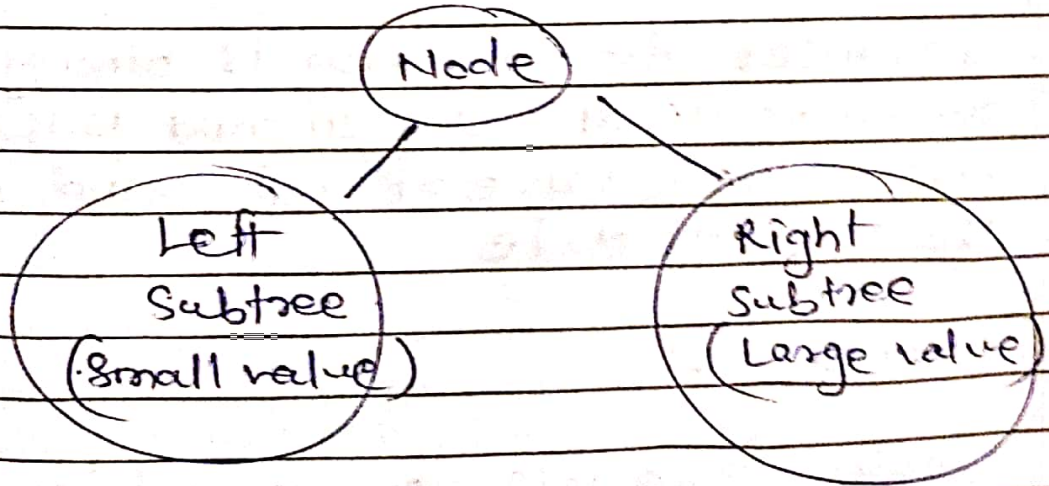
Outcome :

1) Analyze algo & to determine algorithm correctness & time efficiency class
2) Implement abstract datatype (ADT) & data structure for given applicatn
3) Solve problems using algorithmic design techniqu & data structures
4) Analyze of algorithm with respect to time & space complexity.

Theory

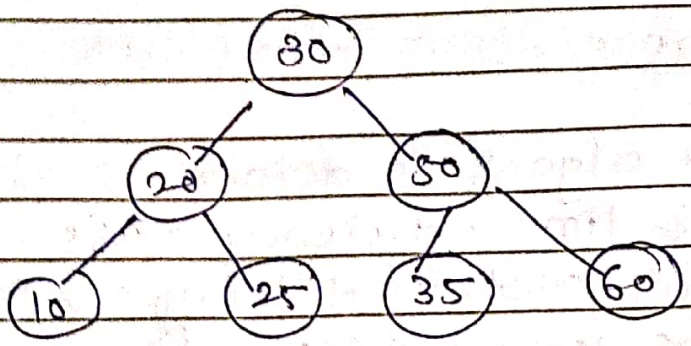
Binary Search tree

A binary tree in which the data of all the nodes in the left sub-tree of the root node is less than the data of the root & data of all nodes in right sub-tree of the root node is more than data root is called BST



Scanned with CamScanner

eg



Applications of BST:

1) BST is used to implement multilevel indexing in database applications
2) BST is also used to implement constructs like dictionary
3) BST can be used to implement various efficiency searching algorithms
4) BST are also used to evaluate the expression using expression tree

ADT BST:

Instances: Binary Search is binary tree in which value of field left child is less than its parent node & value of right child is greater than parent Node

Operations:

1) create: Using this operation a binary search tree can be created

Scanned with CamScanner

Page No.
Date.

2) Display: This operation is used for displaying all nodes of BST

3) Insert: for insertion of any node in BST, this operation is useful

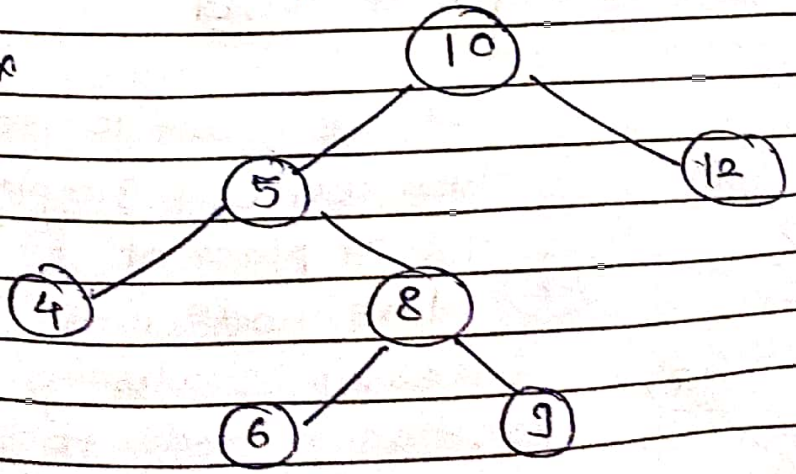
4) Delete: Using this operation, any ~~no~~node from BST can be deleted.

5) Search: This function is used for searching any node from BST.

Search:

1) Node to be searched is called 'key Node'.
2) key node is compared with each node starting from root node if value of key node is greater then search on right subbranch ~~or~~ or on left subbranch
3) If we reach leaf node and still not find the node then "node is not present in tree"

ex



Scanned with CamScanner

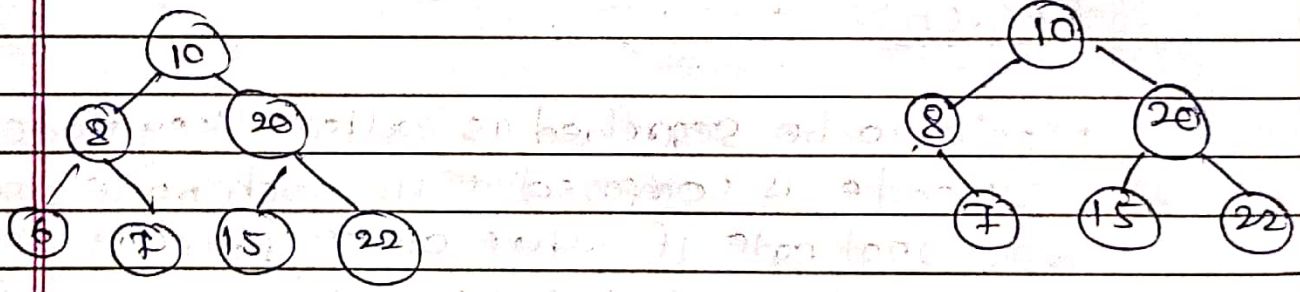
Deletion of element from binary tree :

For deletion of any node from binary Search tree there are three cases.

1) Deletion of leaf node
2) Deletion node having one child
3) Deletion of node having two children.

A) Deletion of leaf Node :

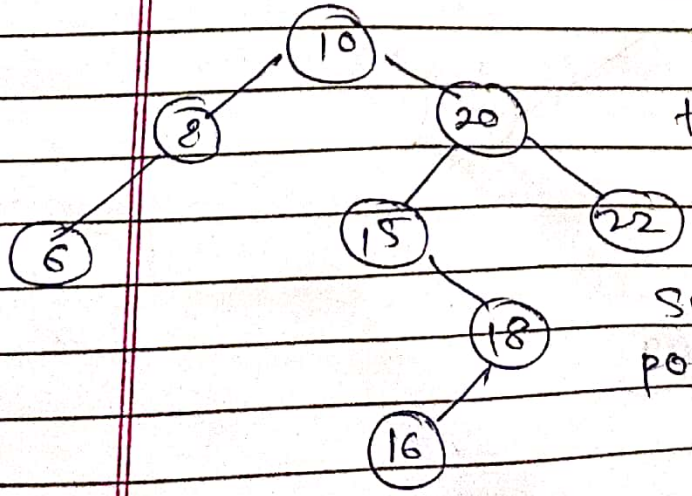
This is the simplest deletion, in which we set left or right pointer of parent node as Null



Before Deletion

After deletion

B) Deletion node having one child :



If we want to delete node 15, then we will simply copy node 18 at place of 15 & then set the node free. The inorder sucessor is always copied at position of node to be deleted

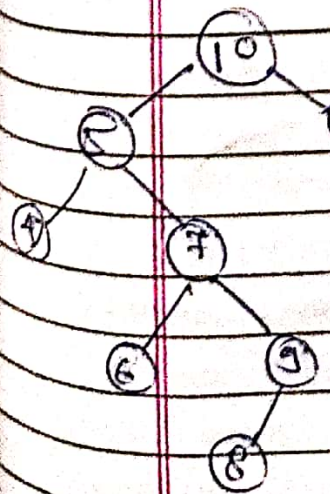
Scanned with CamScanner

step1: find node to be deleted using Search operations
step2: If it has only one child then create a link between its parent & child nodes
step3: Delete the node

c) The node having two children:

Step1: Find node to be deleted using Search operation
Step2: If it has two children, then find largest node in its left subtree
step3: swap both deleting node & node which found in above step
step4: If it comes to case1, then delete using case1 or with case 2
step5: If it comes to then ~~cat~~ check whether deleting node come to case1 or case 2 else goto step2
step6 Repeat same process until node is deleted

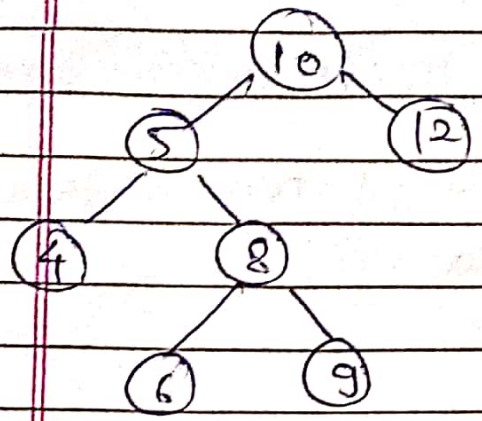
eg



let us consider we want to delete node having value 7 we will find out order successor of node 7. The inorder successor will be simply copied location of node 7

That means copy 8 at position where value of node is 7. Set left pointer of 9 Null

Scanned with CamScanner



Insertion of Node in binary Search :

1) Read the value for node which is to be created and store it in node called *new.
2) Initially if (root != NULL) then root = new then again read
3) If (new -> value < root -> value) then attach new node as left child of root thereoise attach new node as right child of root
4) Repeat above steps for contructing required binary search completely

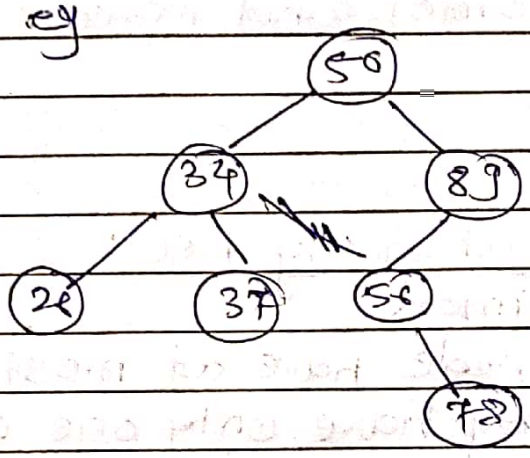
Scanned with CamScanner

Page No.
Date.

Test cases:

1) Random input
2) Sorted input
3) Input for skewed tree concepts

eg



Insert (11) →
Number of comparision = 3

Insert (66) →
Number of comparision = 4

Conclusion:

Binary Search tree is sorted binary tree whose internal nodes each store a key greater than all keys in the left subtree & less than those in right subtree. Using BST, we can perform various operations like searching, deleting, inserting effectively

Scanned with CamScanner