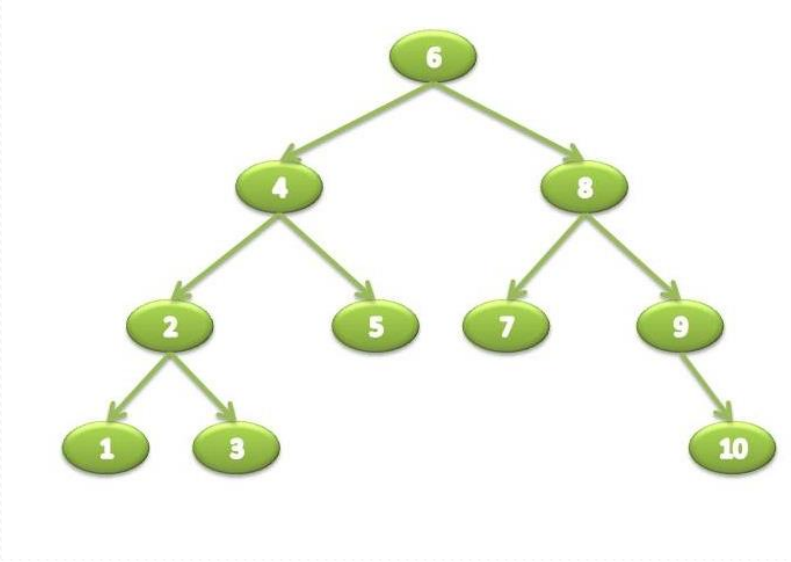
Binary Search Trees

Computer Representation



Data Structures and Algorithms

Binary Search Tree

Create a BST for:

10, 30, 50, 15, 10, 30, 17, 35, 25, 70

Binary Search Tree

- Properties
- Packing the Node
- Insertion Operation
- Traversals
- Deletion Operation

Properties

- The left subtree of a node contains keys less than the node's key.
- The right subtree of a node contains keys greater than or equal to the node's key.
- The left and right subtree must each also be a binary search tree.

Packing the Node

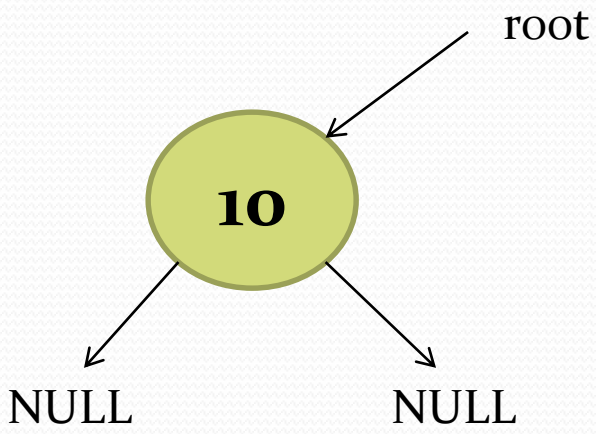
```
struct tree
{
    int data;
    struct tree * left;
    struct tree * right;
};
typedef struct tree TREE;
```

Insertion Operation

TREE * root = NULL

Insertion Contd..,

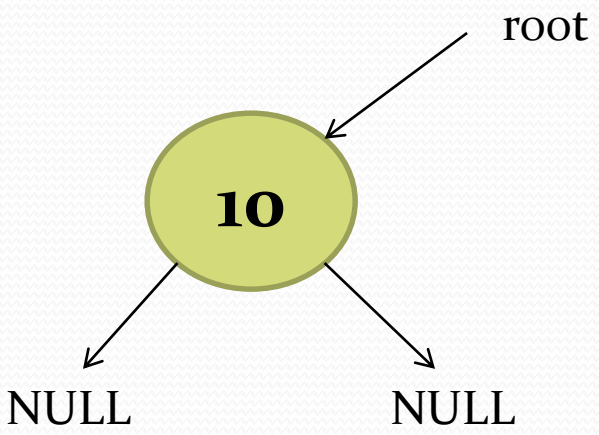
The first node walks in:

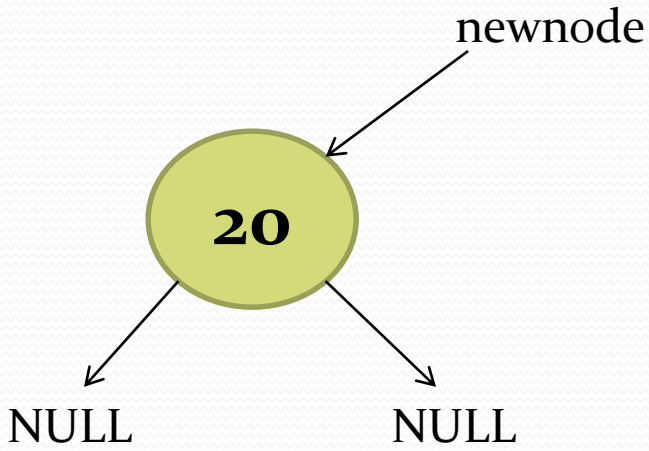


… and becomes the root node

Insertion Contd..,

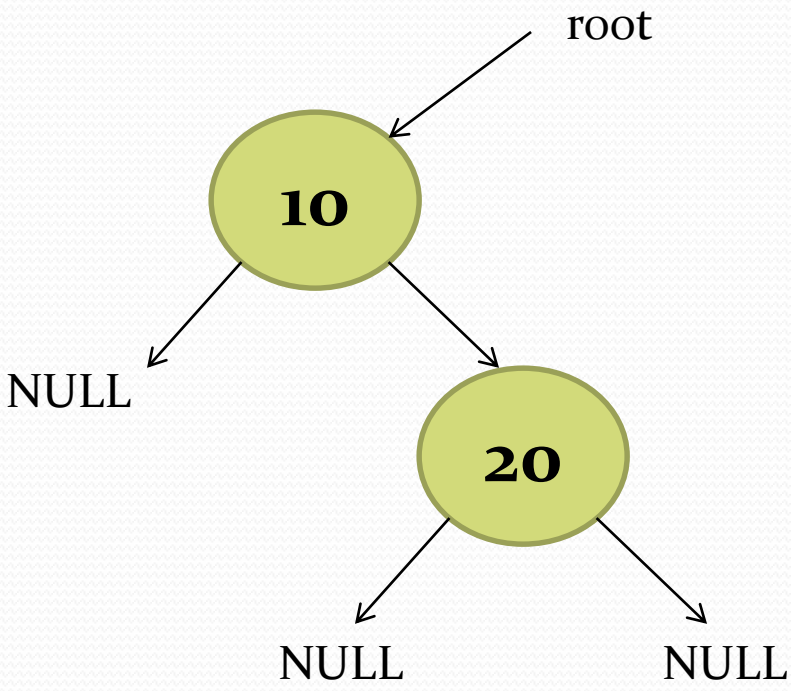
The second node walks in:





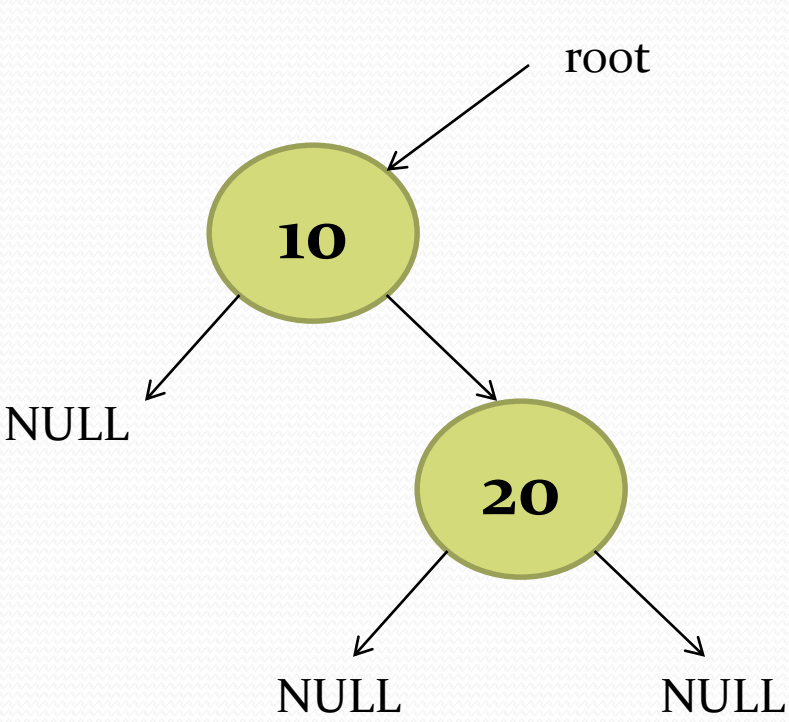
Insertion Contd..,

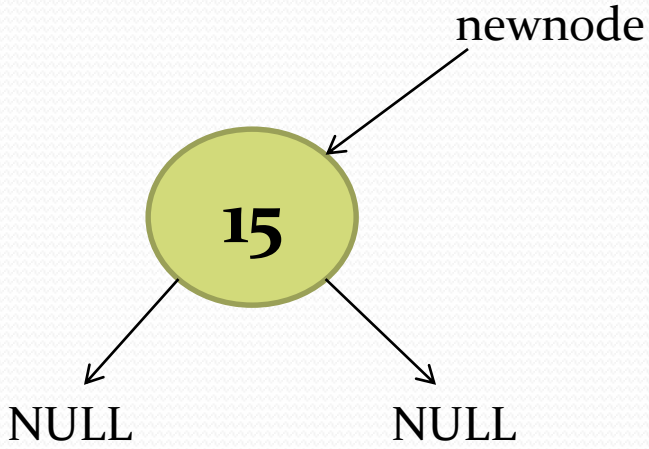
When the second node walks in:



Insertion Contd..,

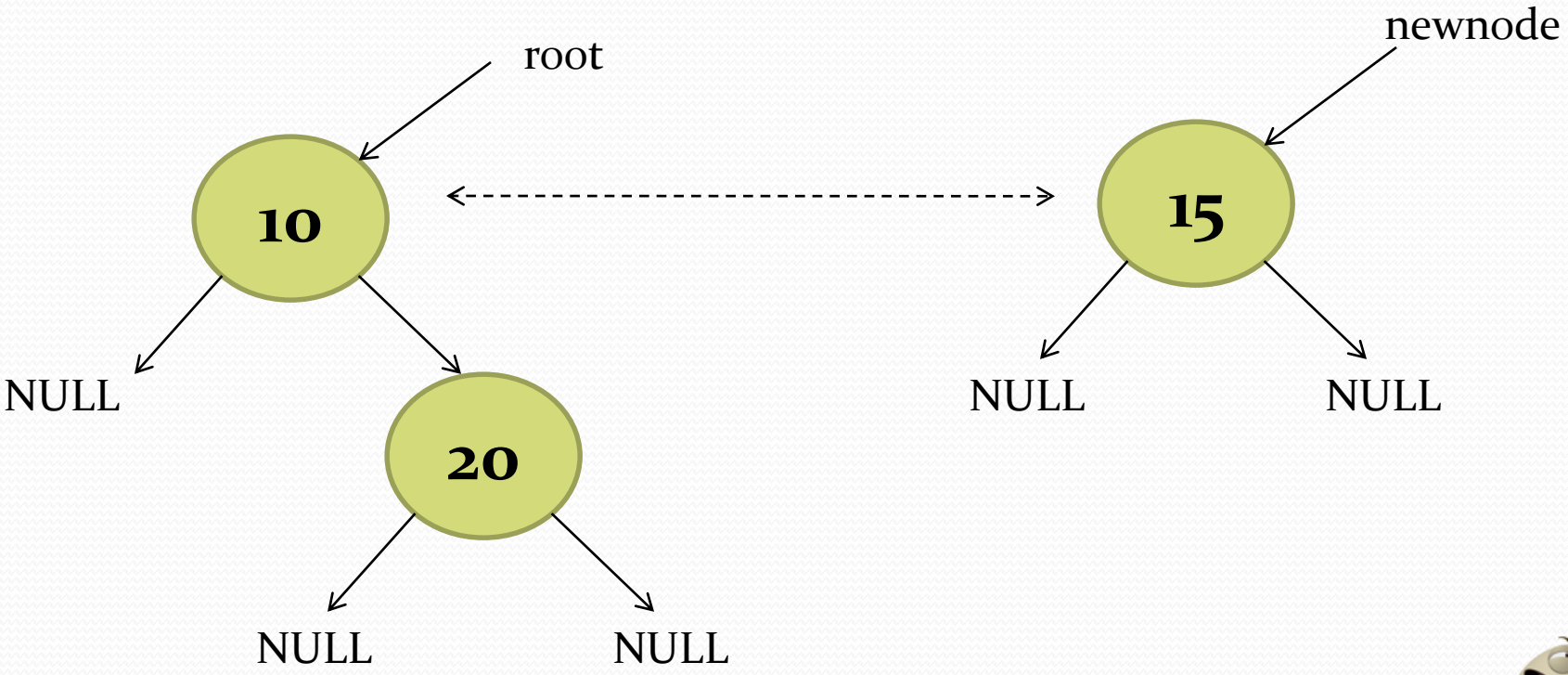
Yet another node walks in:





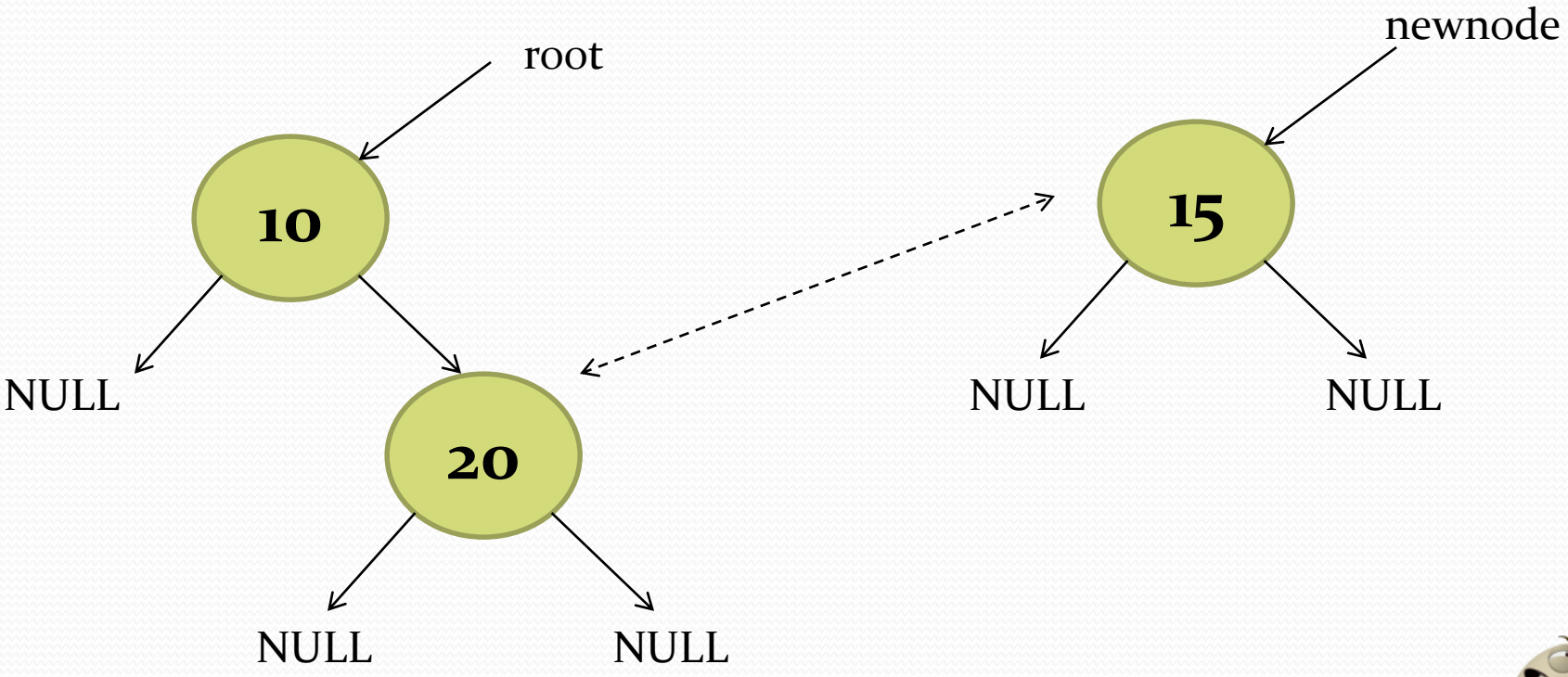
Insertion Contd..,

Yet another node walks in:



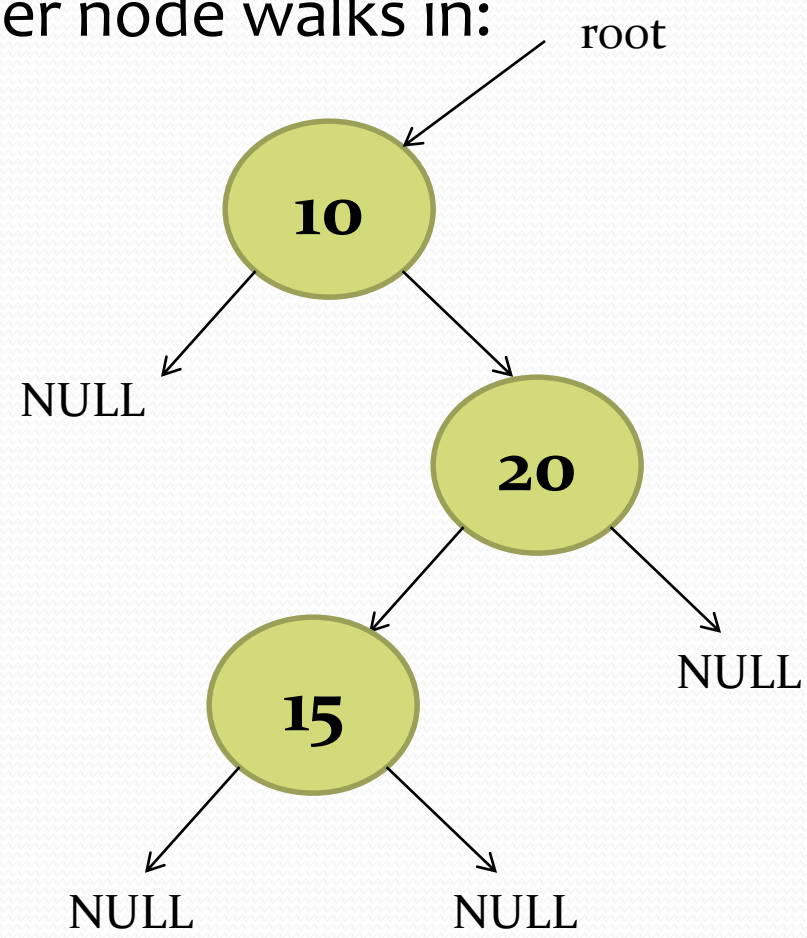
Insertion Contd..,

Yet another node walks in:



Insertion Contd..,

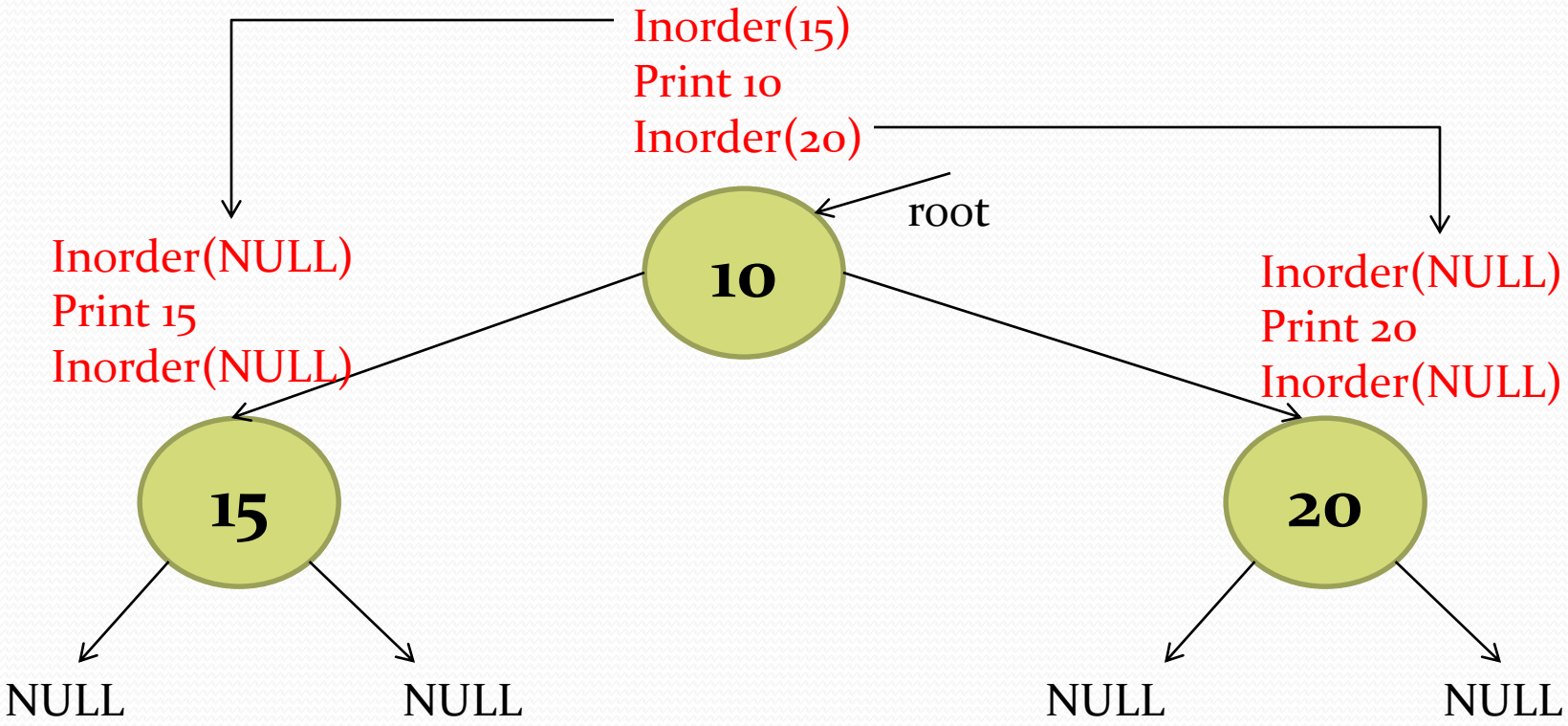
Yet another node walks in:



Tree Traversals

- In-order
  - Left, Root, Right
- Pre-order
  - Root, Left, Right
- Post-order
  - Left, Right, Root

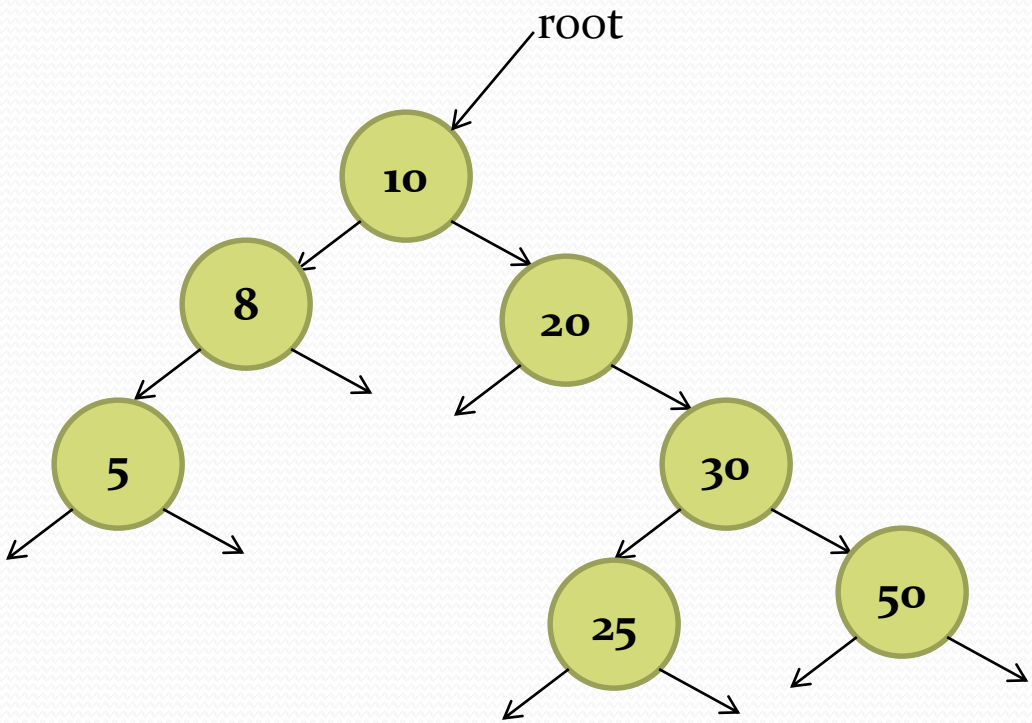
Recursive Inorder:



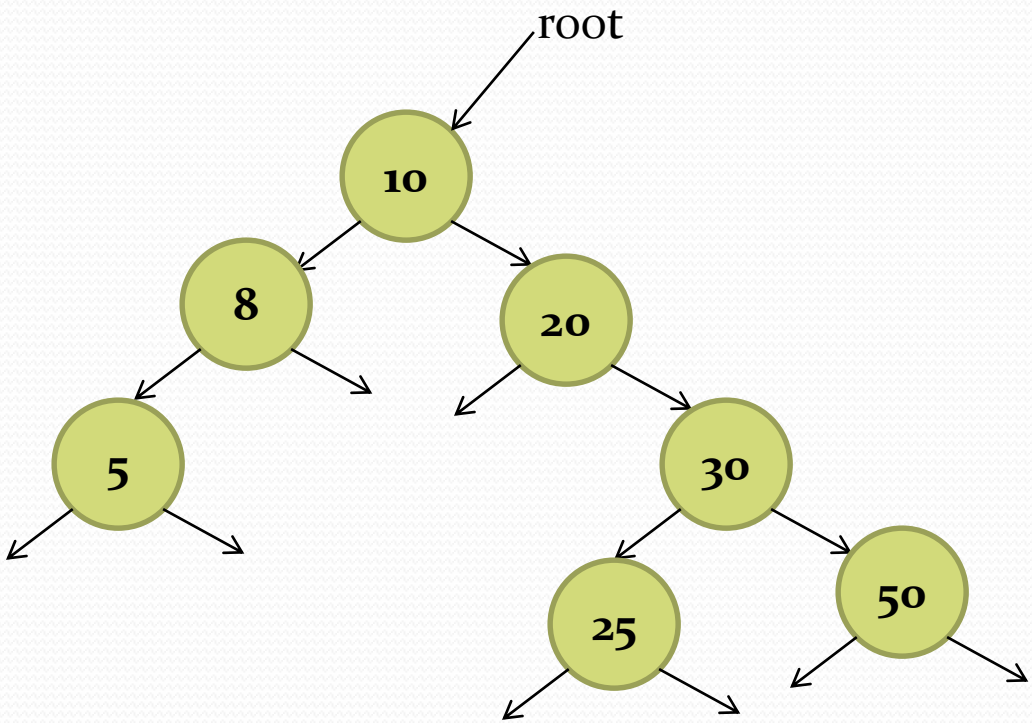
Deleting a node

- Node may not be present
- A leaf node
- A non leaf node with 1 child
- A non leaf node with 2 children
- Root node

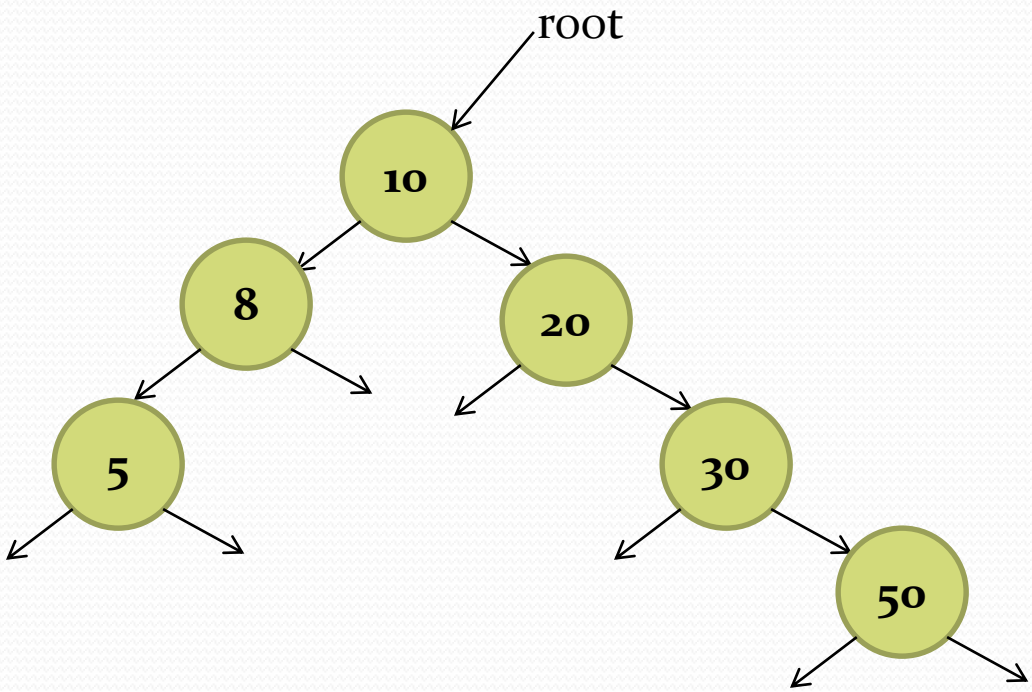
Delete: 60



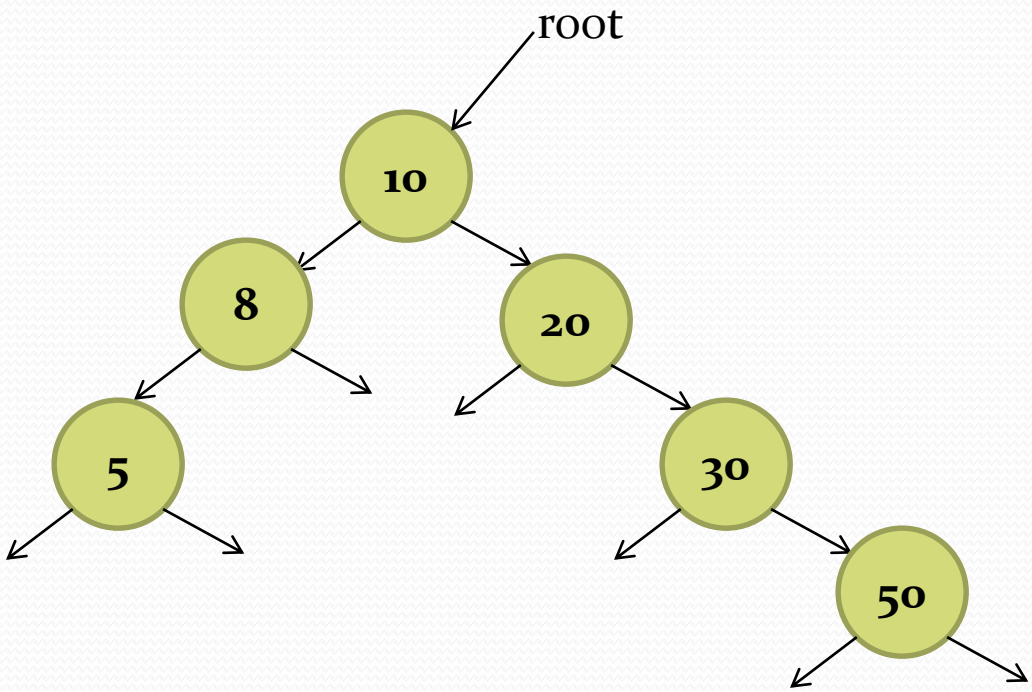
Delete: 25



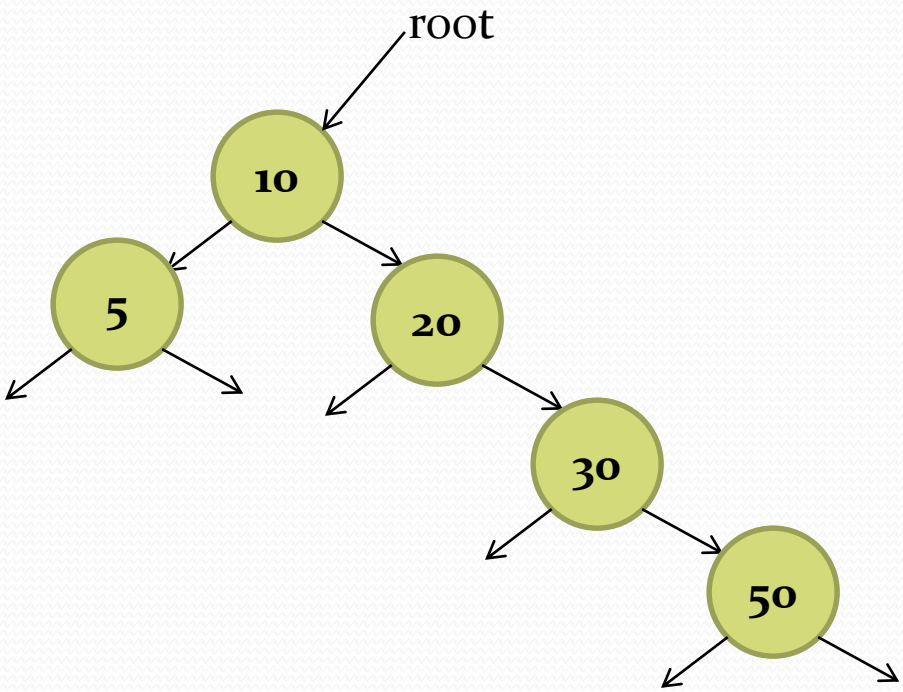
Deleted: 25



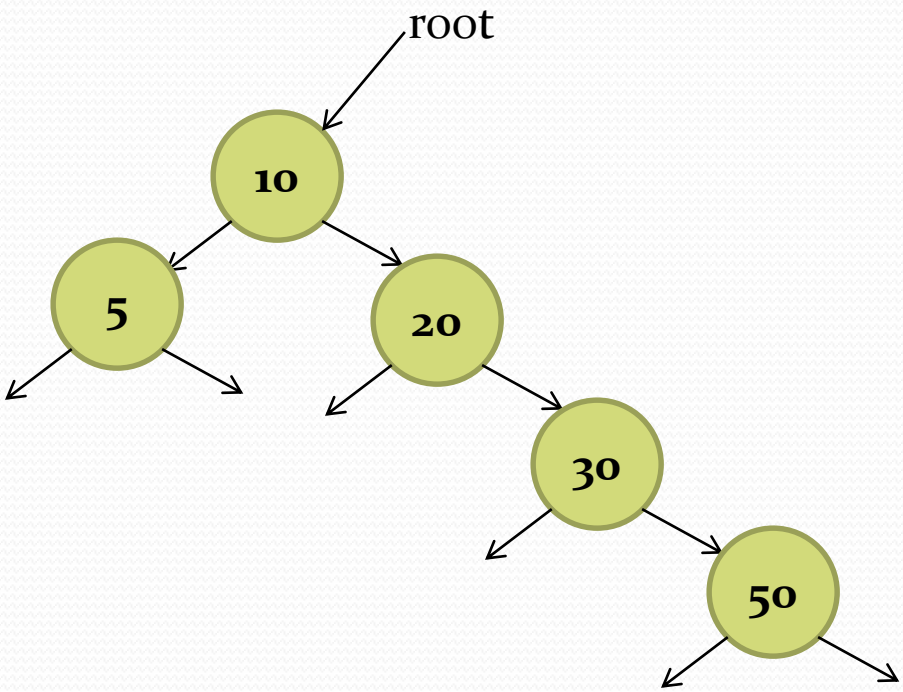
Delete: 8



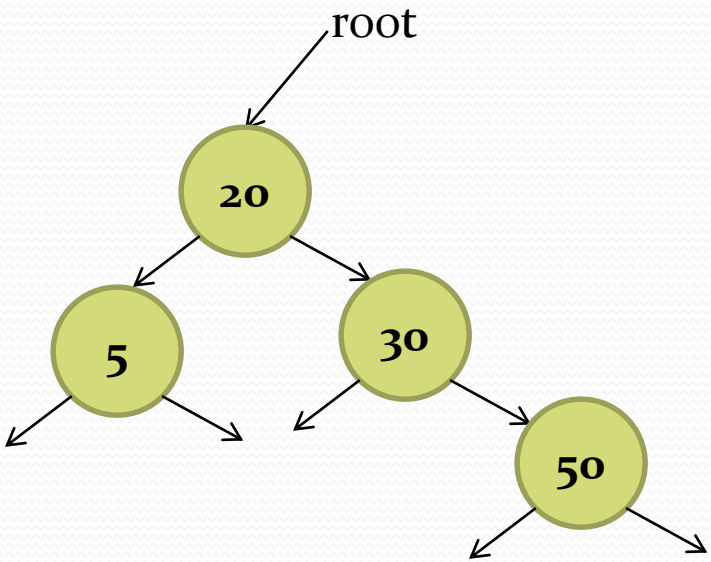
Deleted: 8



Delete: 10



Deleted: 10



Thank you.