# DSA through C++

# Binary_Search_Tree


**Mohammad Tasin (Tasin Coder)**

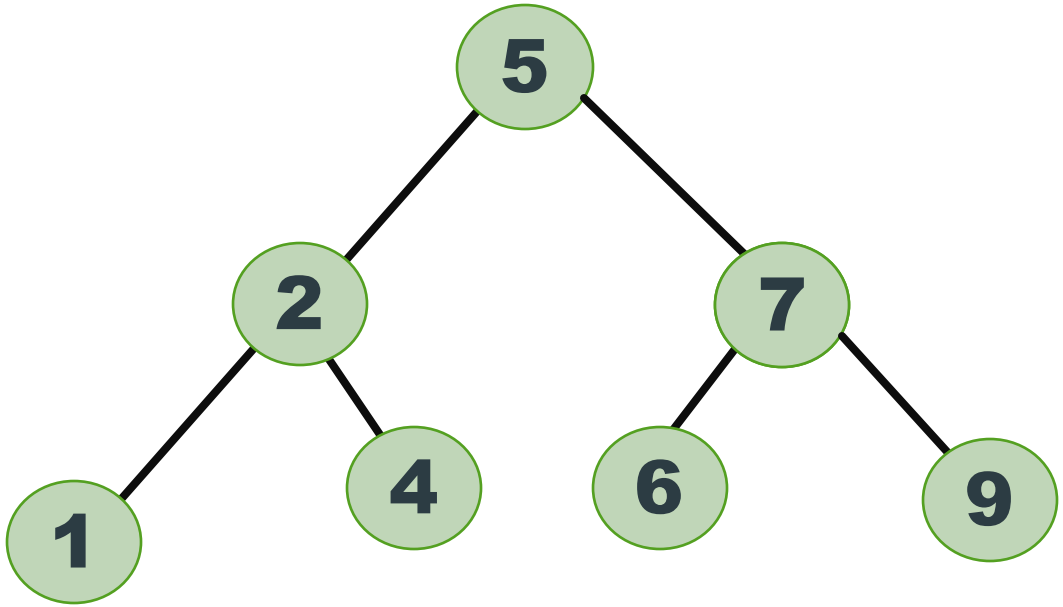
# Agenda

- Binary Search Tree
- Implementation of BST

# Binary Tree

- A binary search tree is the most important data structure, that enables one to search for and find an element with an average running time $f(n) = O(\log_2 n)$

- Duplicate values are not allowed in BST ( By default )


5
2
7
1
4
6
9

- **Binary Search Tree is a binary tree with the value at node N is Less than every value in the left subtree of N and is greater than every value in the right subtree of N.**
- **Unless, explicitly said, BST doesn't allow duplicate values**

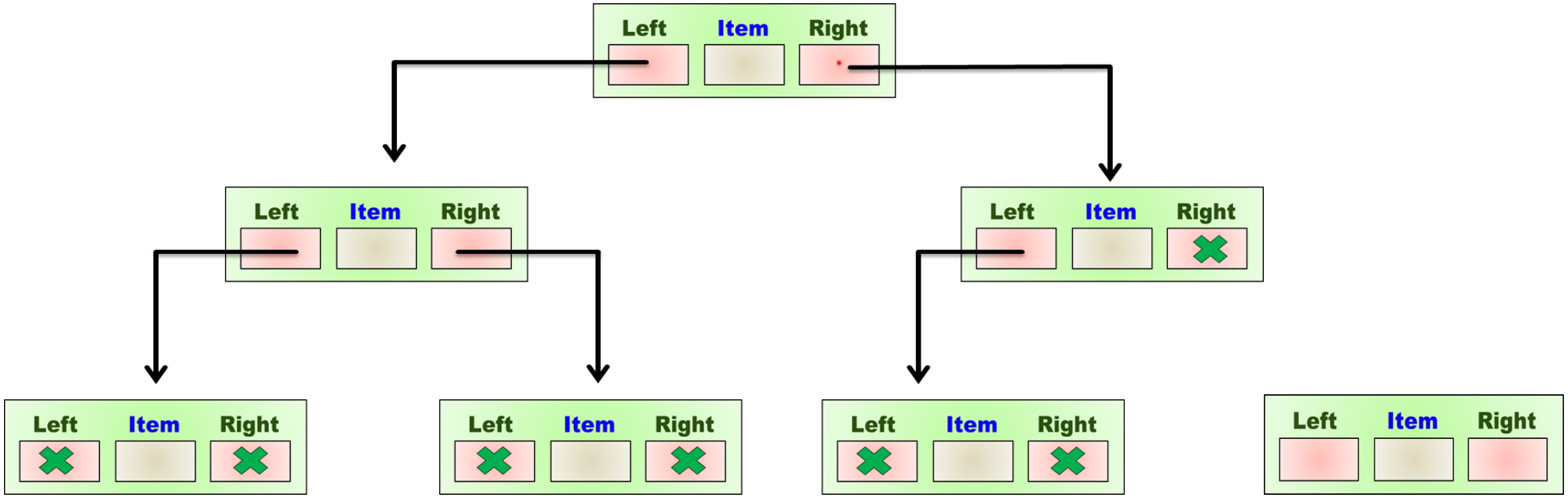
# Implementation of BST

- node
- Insertion
- Traversing
- Search
- Deletion

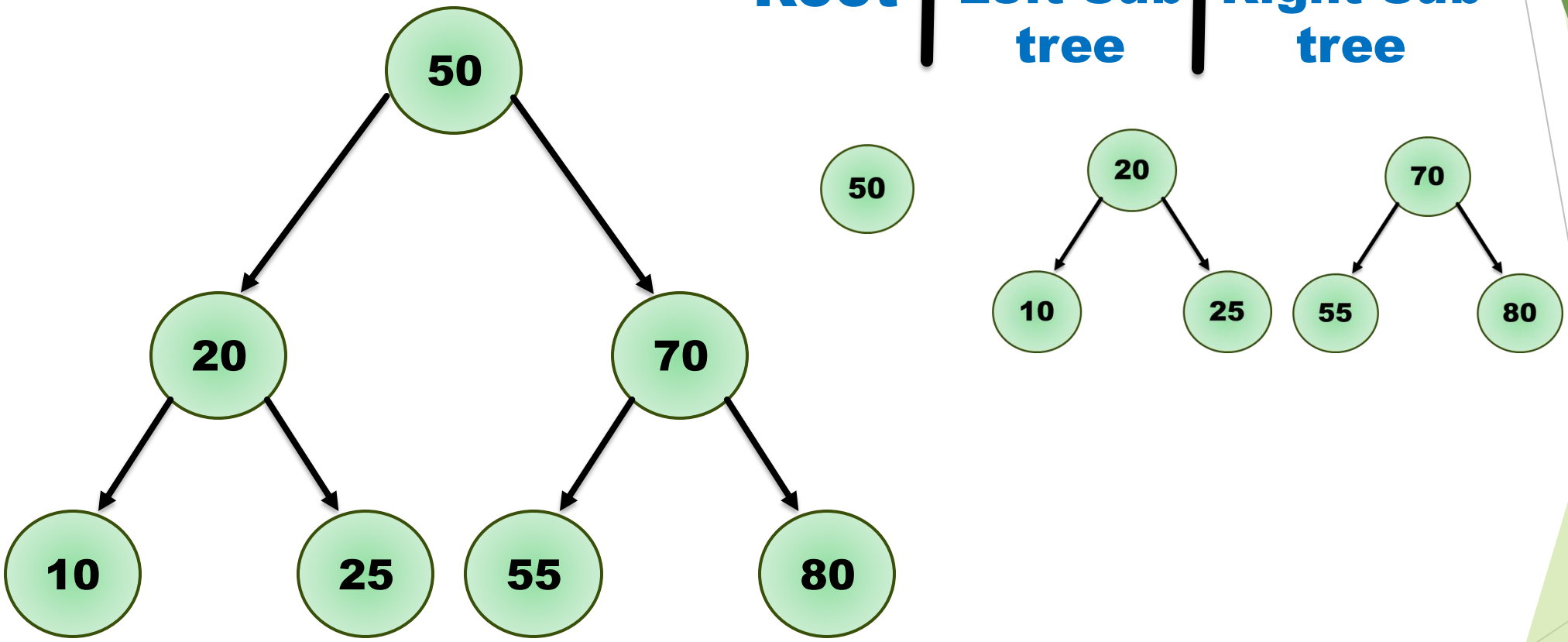
# Node

```
struct Node
{
    Node *Left;
    int Item;
    Node *Right;
};
```

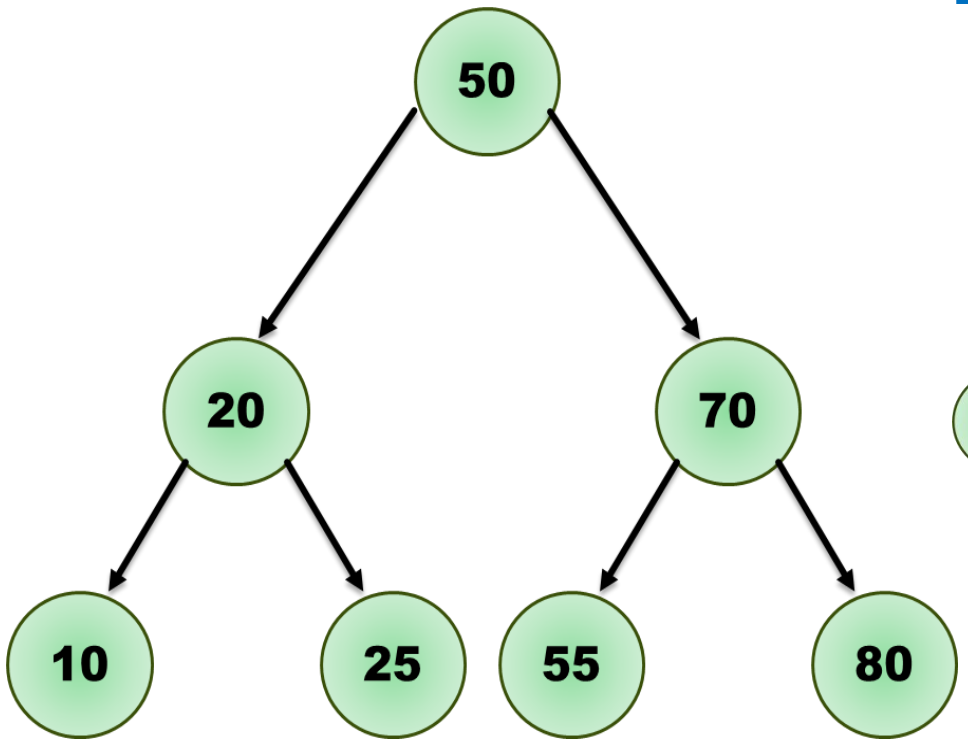
# Insertion


Left
Item
Right
Left
Item
Right
Left
Item
Right
Left
Item
Right
Left
Item
Right
Left
Item
Right
Left
Item
Right

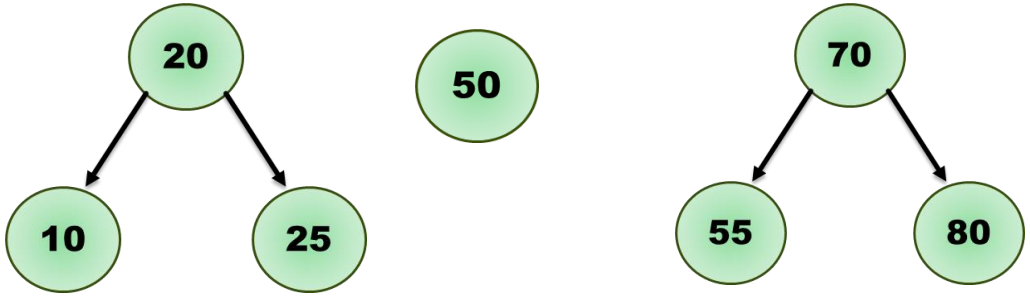
# Preorder Traversing



# Inorder Traversing

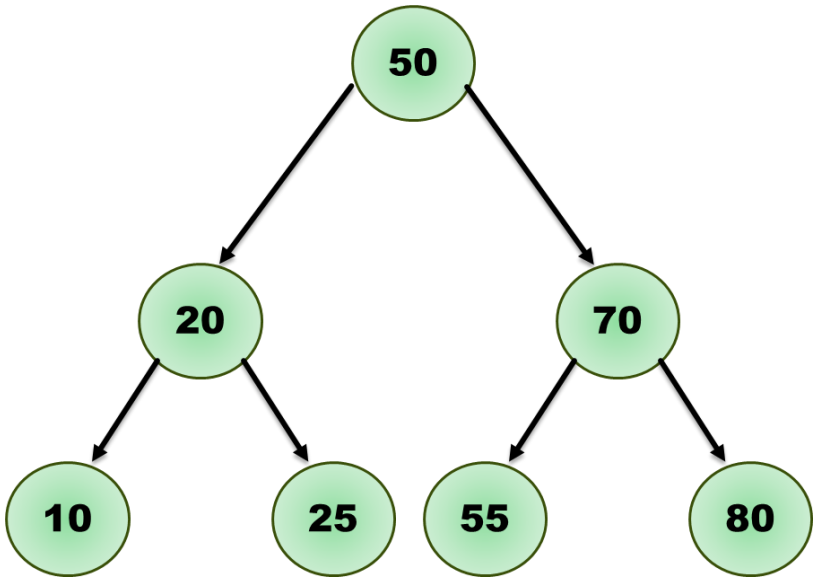



**Left Sub tree** | **Root** | **Right Sub tree**

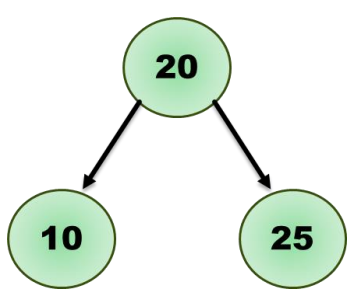


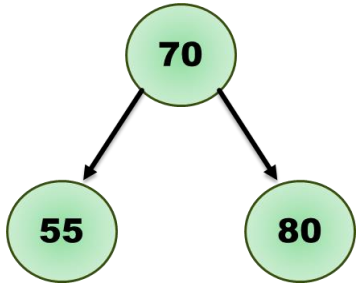

- **Inorder traversing of BST always gives sorted order of elements**

# Postorder Traversing




**Left Sub tree** | **Right Sub tree** | **Root**




# Deletion