DSA using Python

Binary Search Tree

Saurabh Shukla (MySirG)

Agenda

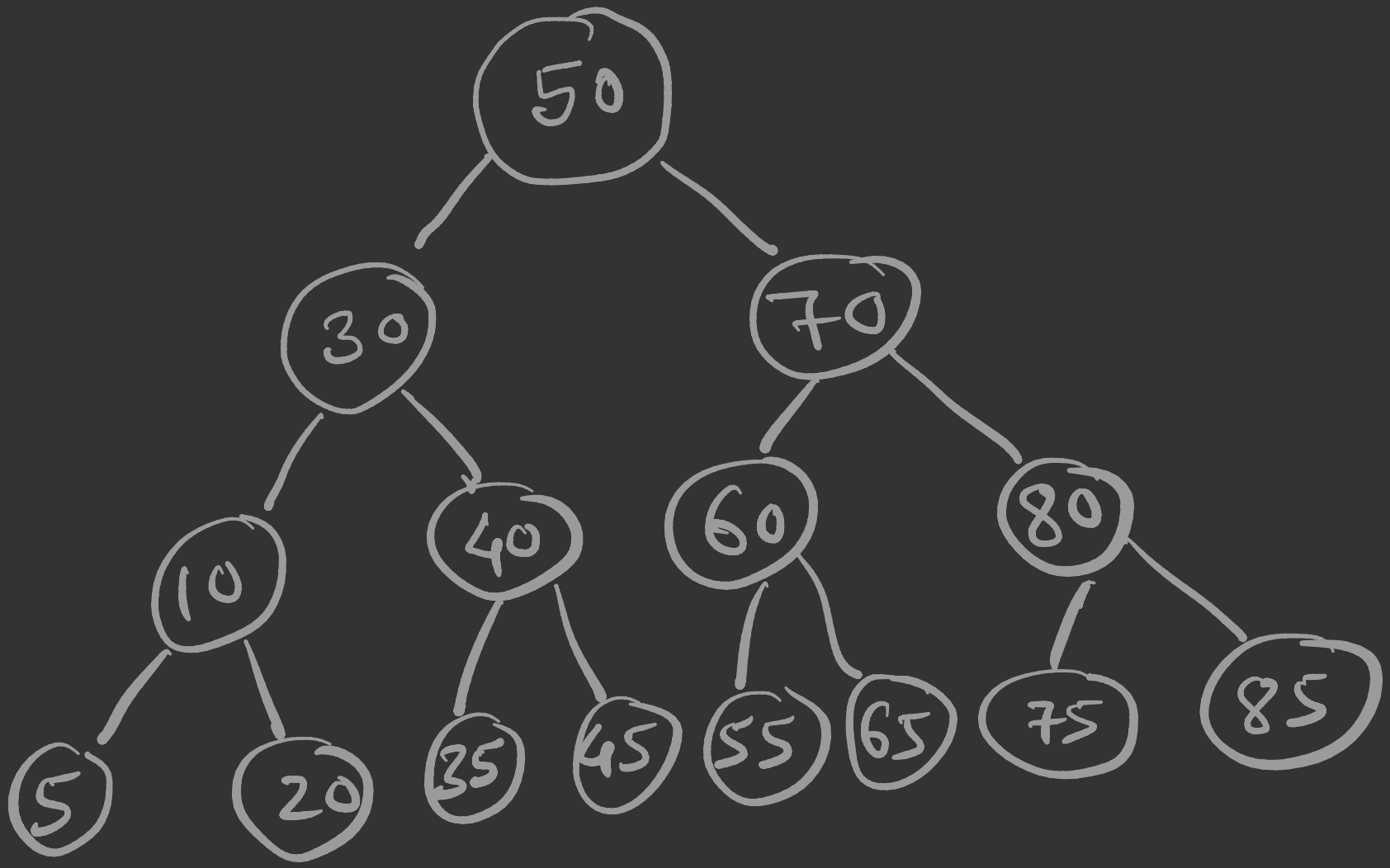
1. Binary Search Tree
2. Implementation of BST

Binary Search 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)


50
30
70
10
40
60
80
5
20
35
45
55
65
75
85

Binary Search Tree is a binary tree with the value at node N is greater than every value in the left subtree of N and is less than every value in the right subtree of N.

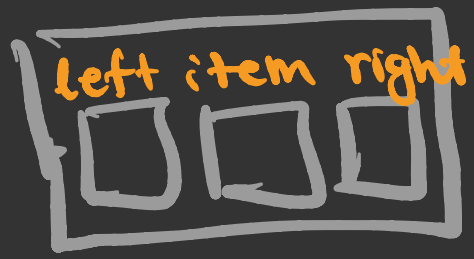
Unless, explicitly said, BST doesn't allow duplicate values.

Implementation

1. node
2. Insertion
3. Traversing
4. Search
5. Deletion

node

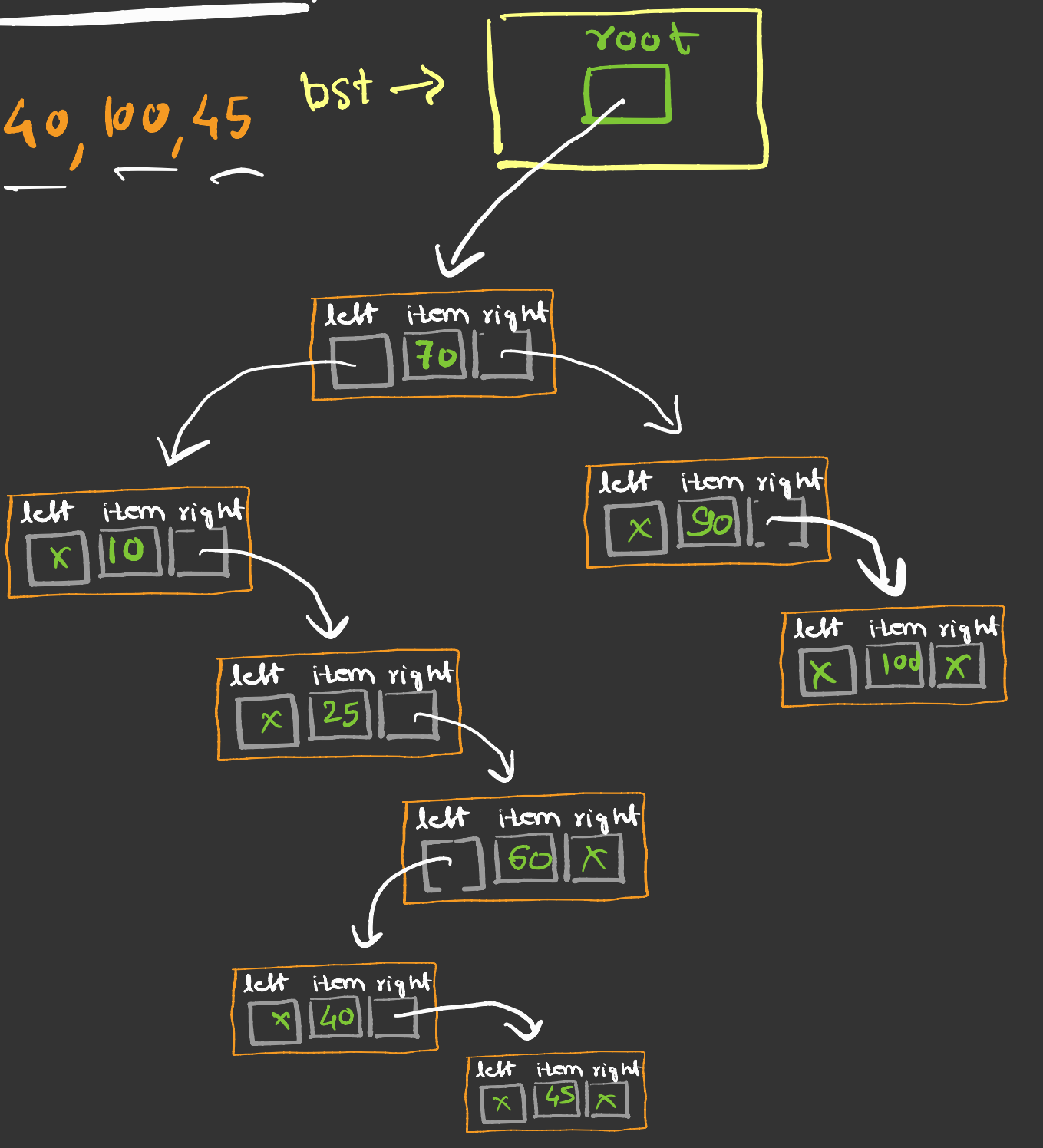
class Node :



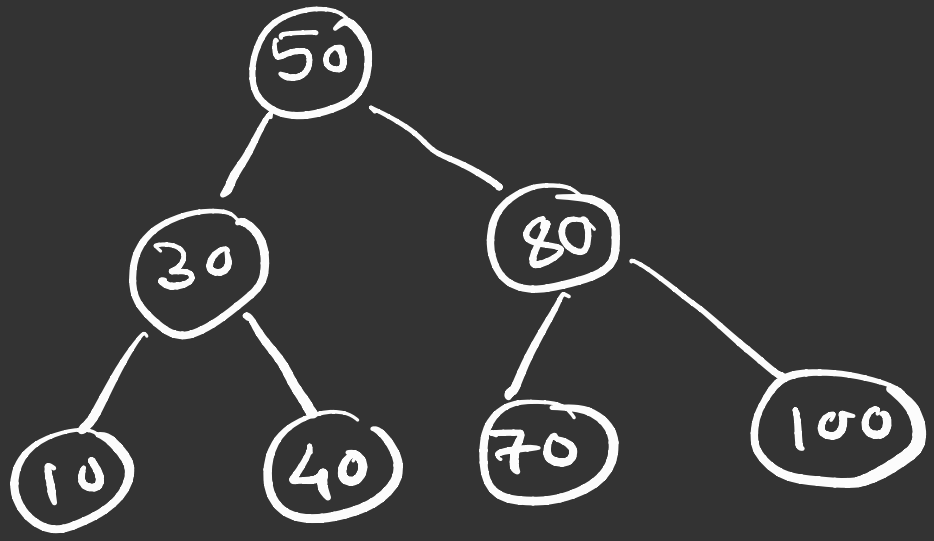
Insertion

bst = BST()

70, 10, 25, 90, 60, 40, 100, 45

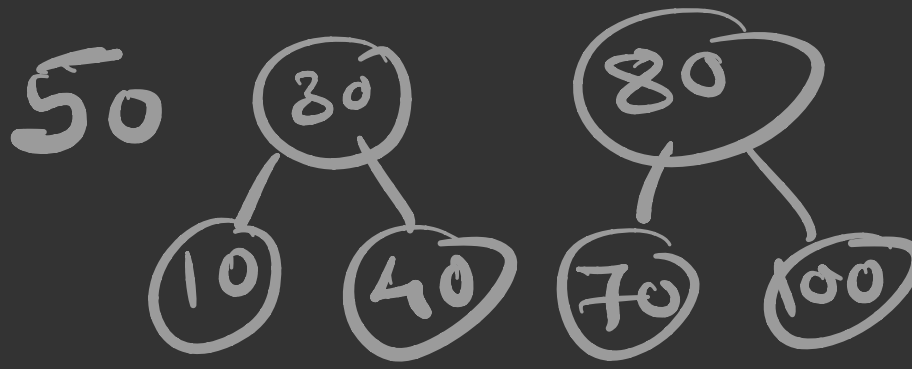


traversing



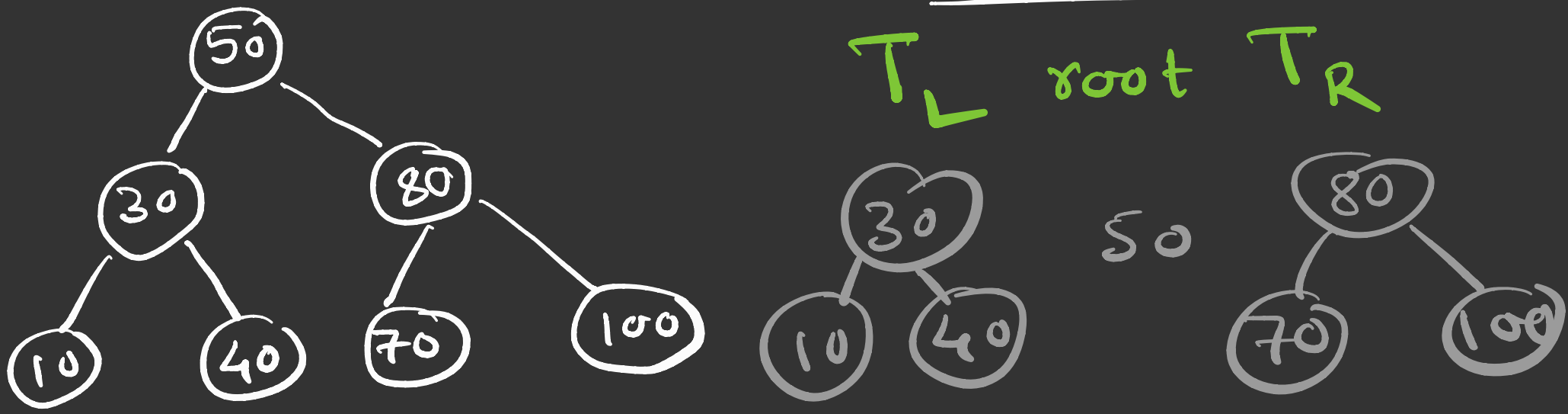
preorder

root T_L T_R



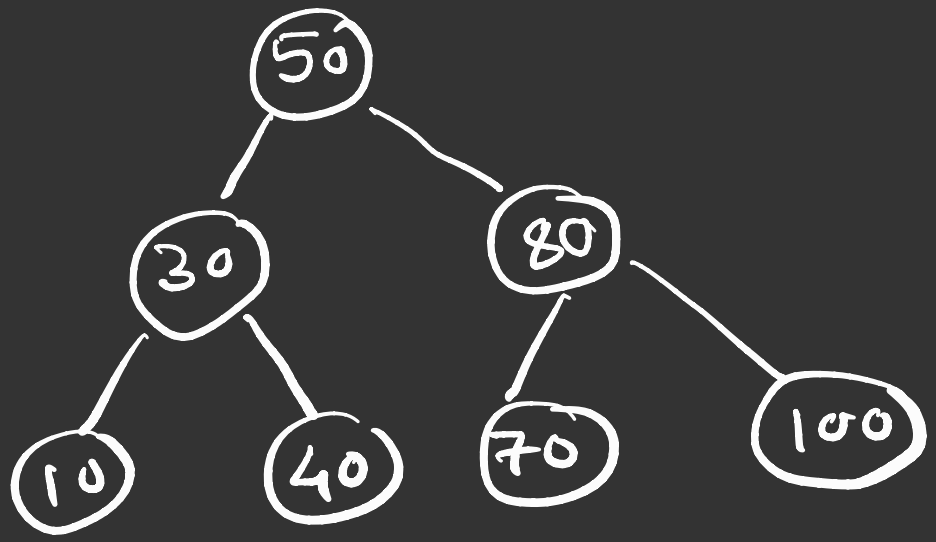
50 30 10 40 80 70 100

Inorder



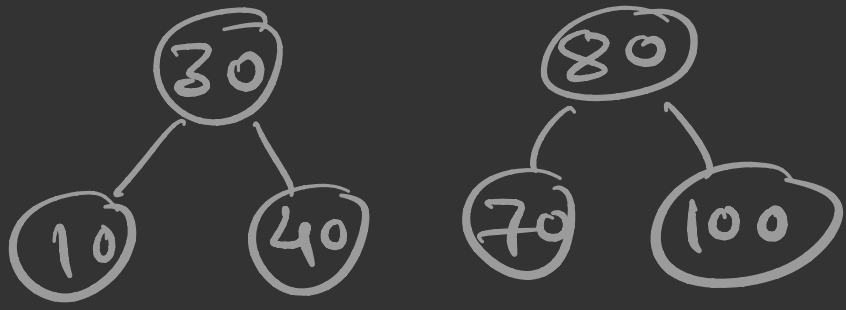
10 30 40 50 70 80 100

Inorder traversing of BST gives values in sorted order



Postorder

T_L T_R root



10 40 30 70 100 80 50

Deletion

1. No child (10)
2. Single child (40)
3. Two children (80)