DSA through Java

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

$$f(n) = O(\log_2 n)$$

$\log_2 1023 \cong 10$

L_0

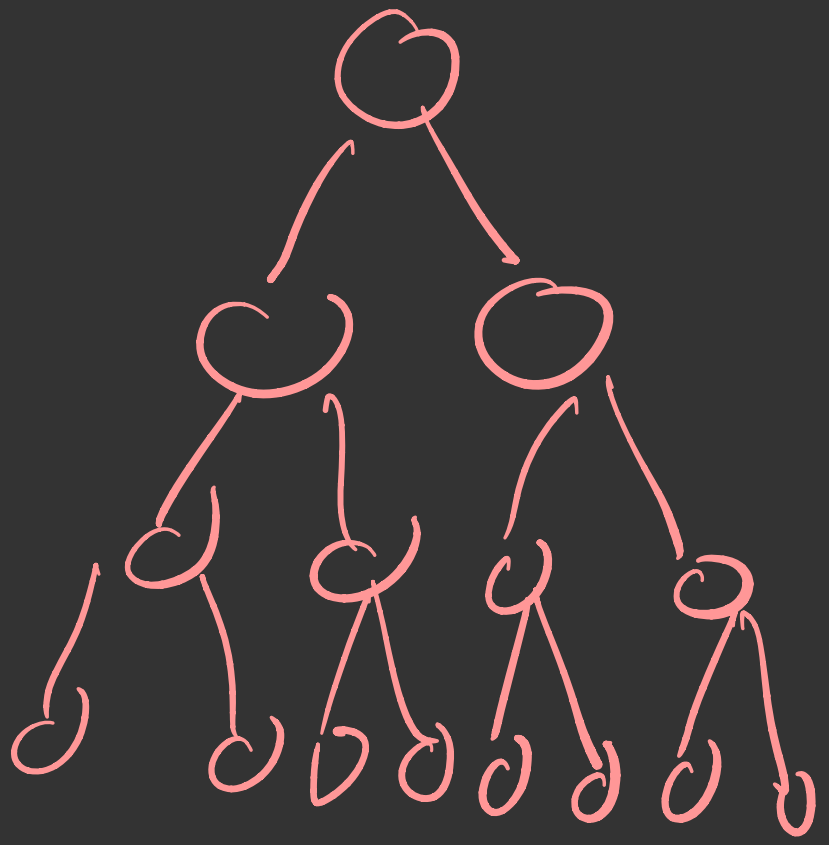
L_1

L_2

L_3

$\vdots$

L_9



$1+2+4+8+\ldots+512$

$= 1023$

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

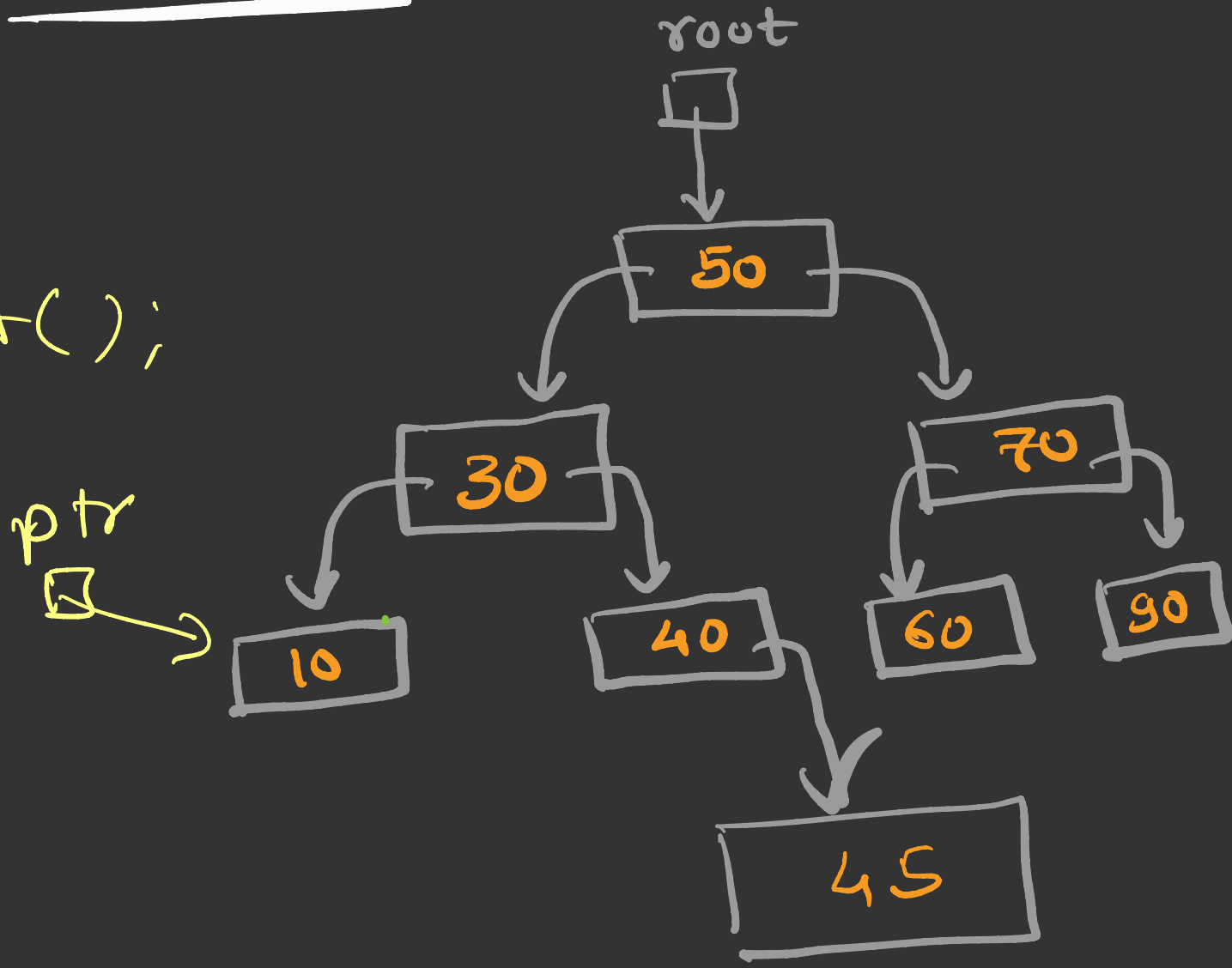
```
class Node
{
    private Node left;
    private int item
    private Node right;

    getters & setters

}
```

Insertion


```
ptr = ptr.getLeft();
```

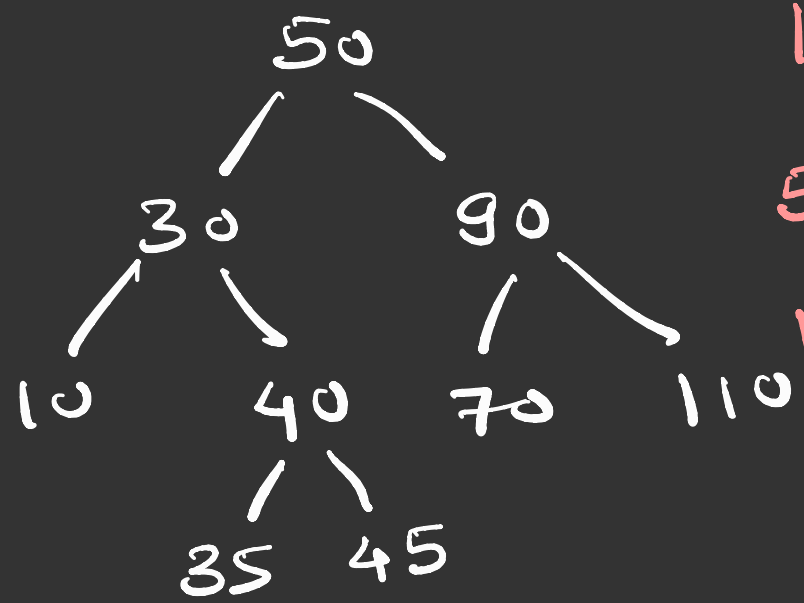


Traversing

Inorder — Left ST Root Right ST

preorder — Root Left ST Right ST

postorder — Left ST Right ST Root



10, 30, 35, 40, 45, 50, 70, 90, 110

50 30 10 40 35 45 90 70 110

10 35 45 40 30 70 110 90 50