

DSA using Python

Binary Tree



Saurabh Shukla (MySirG)

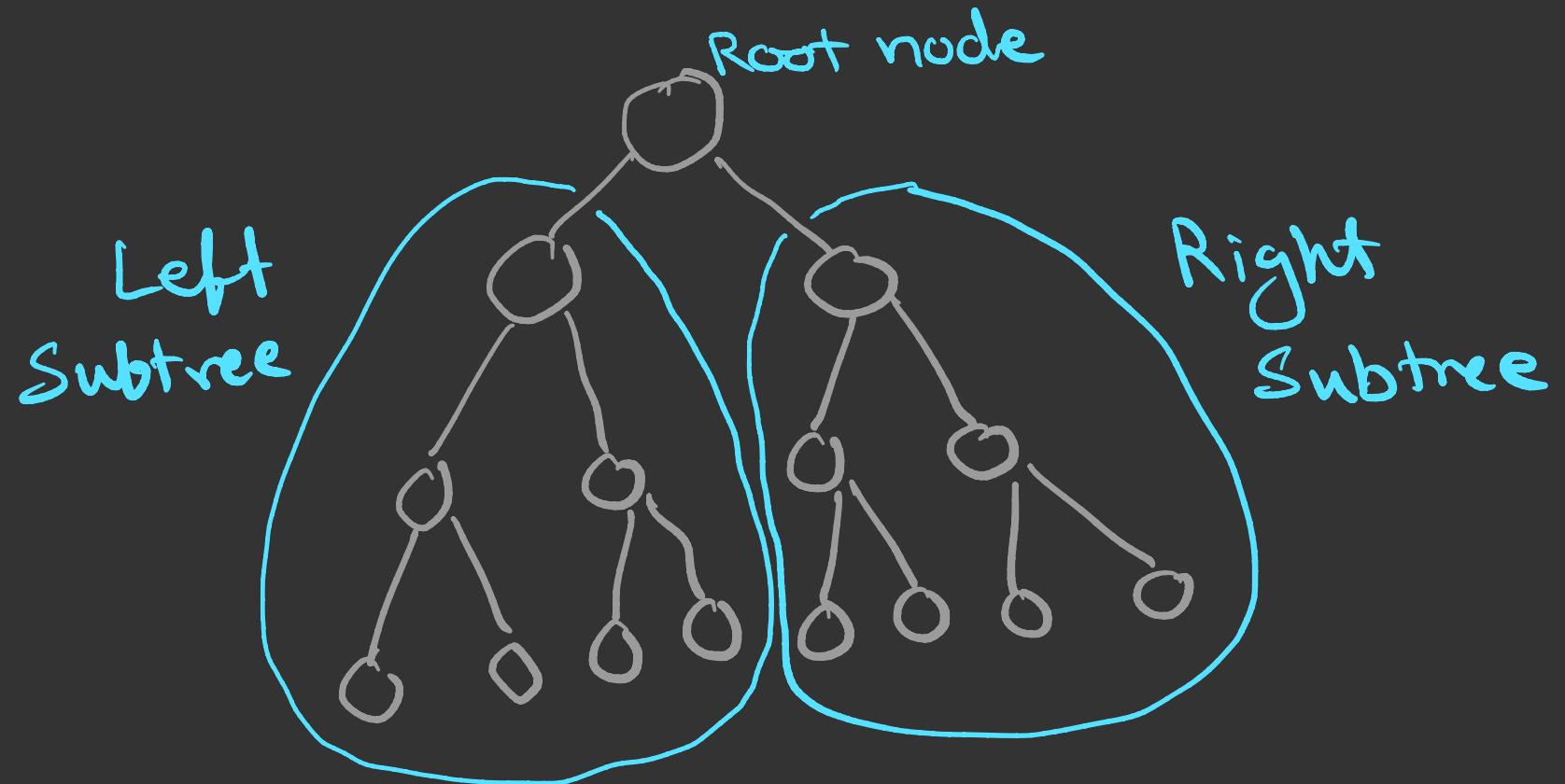
Agenda

- ① Binary Tree
- ② Complete Binary Tree
- ③ Almost Complete Binary tree
- ④ Strict Binary Tree
- ⑤ Representation of Binary tree

Binary Tree

A binary tree is defined as a finite set of elements, called nodes, such that

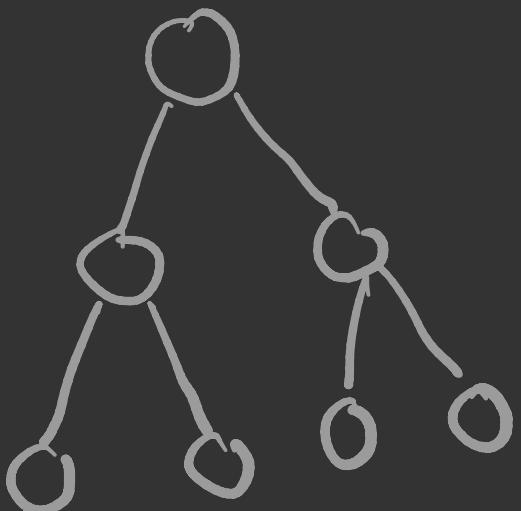
- T is empty (called the Null tree or empty tree), or
- T contains a distinguished node R , called the root of T , and the remaining nodes of T form an ordered pair of disjoint binary trees T_1 and T_2



Any node in the binary tree has either
0, 1 or 2 child nodes.

Complete Binary Tree

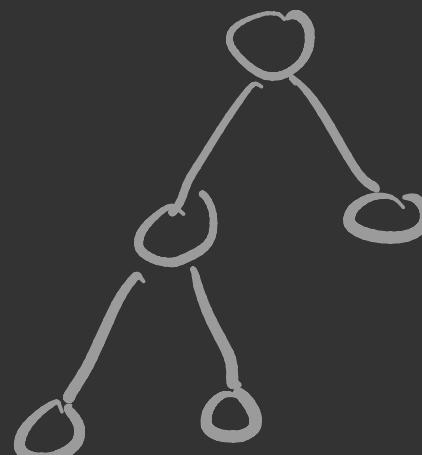
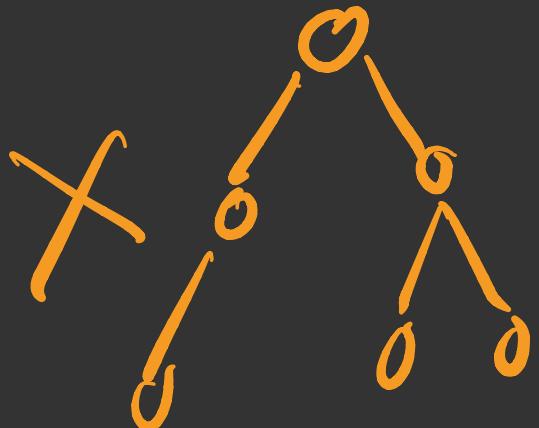
All levels are completely filled.



L_0	-	1
L_1	-	2
L_2	-	4
L_3	-	8
L_4	-	16
L_5	-	32
L_6	-	64
L_7	-	128
L_n	-	2^n

Almost Complete Binary Tree

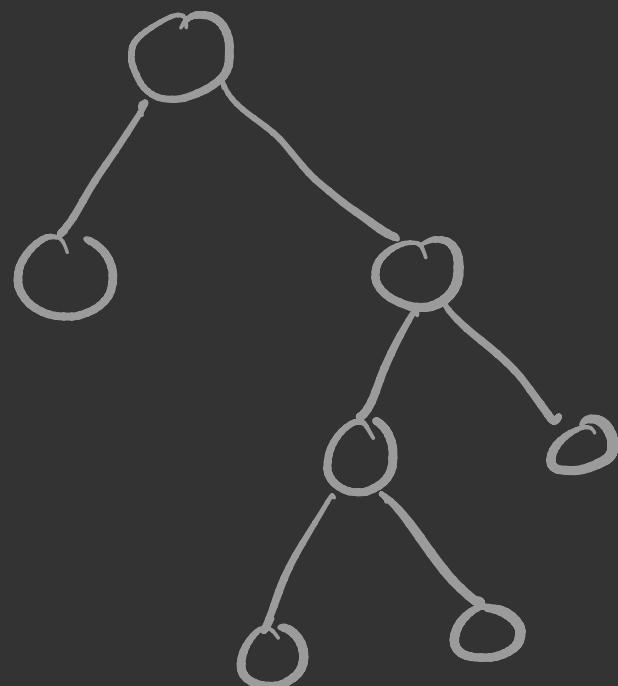
All levels are completely filled, except possibly the last level and nodes in the last level are all left aligned.



Strict Binary Tree

Each node of a strict Binary Tree will have either 0 or 2 children.

Full Binary Tree



Representation of Binary Tree

There are two possible representations of binary tree

- ① Array Representation
- ② Linked Representation (by default)

Linked Representation

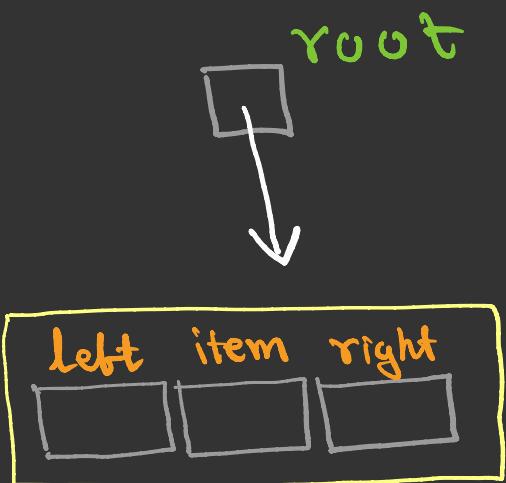


node
====

class Node :

```
def __init__(self, item=None, left=None, right=None):  
    self.item = item  
    self.left = left  
    self.right = right
```

Root



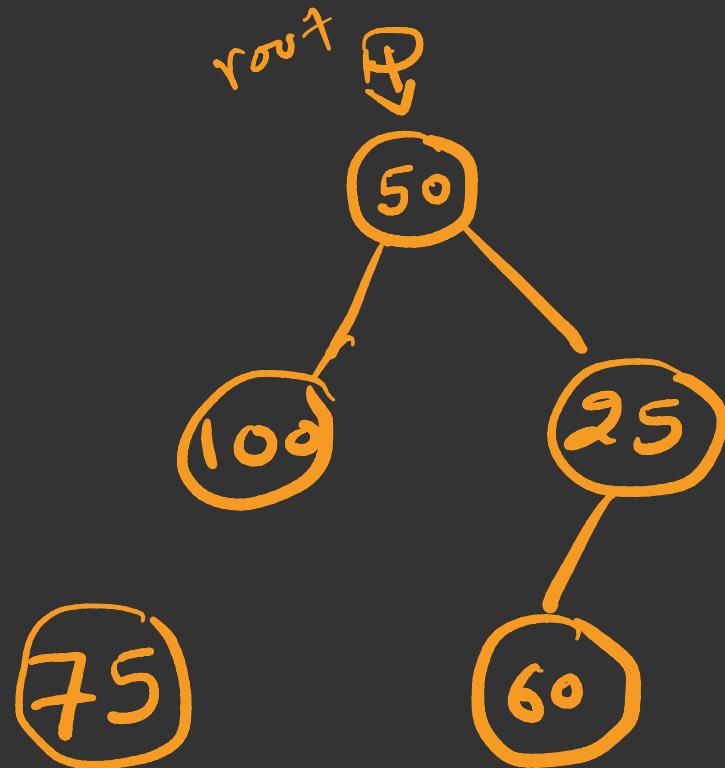
- . root is a node pointer
- when root contains NULL , tree is empty.

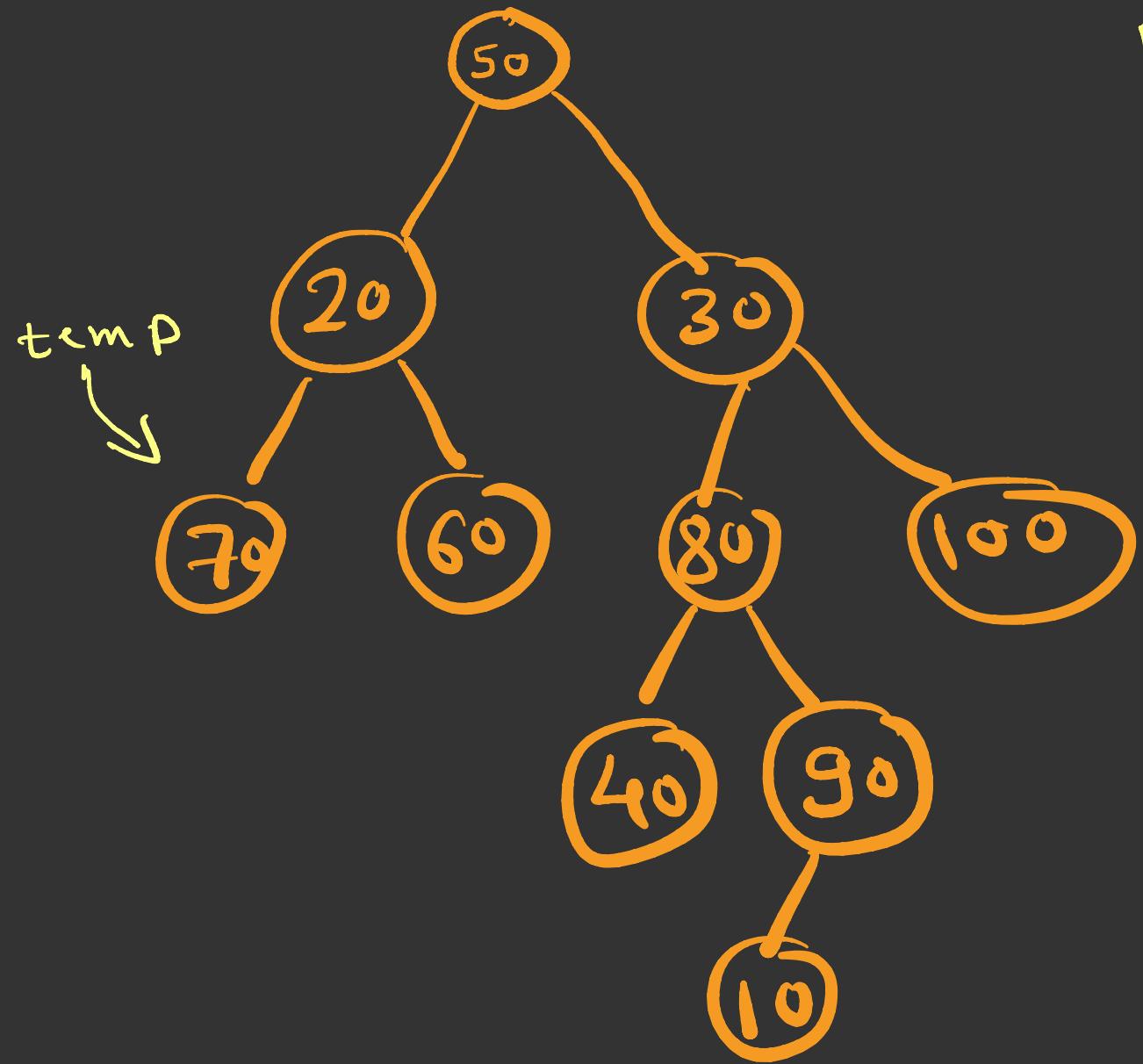
root



Discuss

- How to insert an item in a BT?
- How to traverse a BT?





print (temp.item) 50
temp = temp.left
print (temp.item) 20
temp = temp.left