Data Science & AI
&
NIC - Param

Python-For Data Science

Binary Tree

Lecture No.- 01

By- Pankaj Sharma Sir

Recap of Previous Lecture

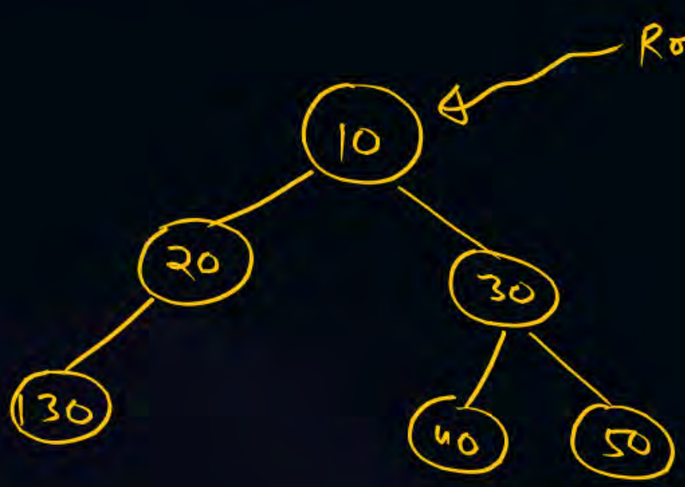
Topic Stack and Queues Part-02

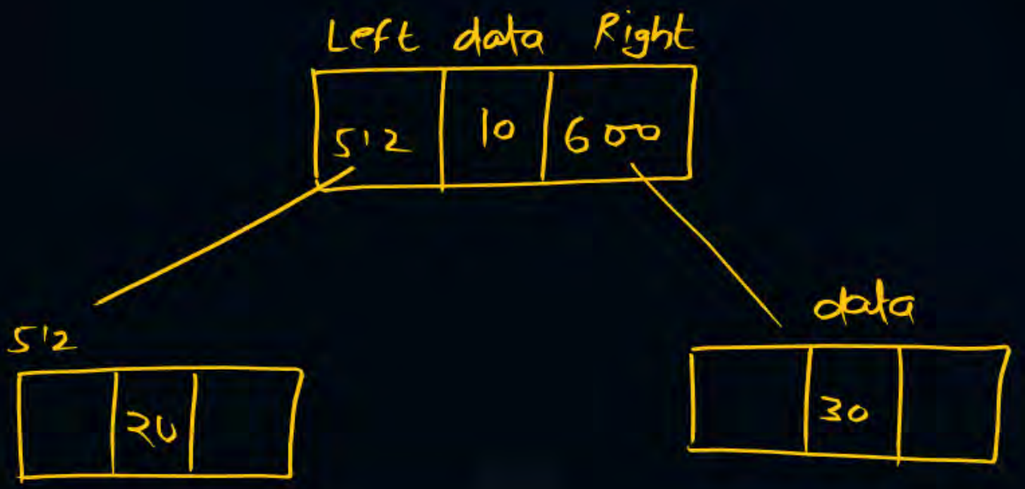
Topics to be Covered

Topic Trees Part 01

Topic : Trees

Binary tree

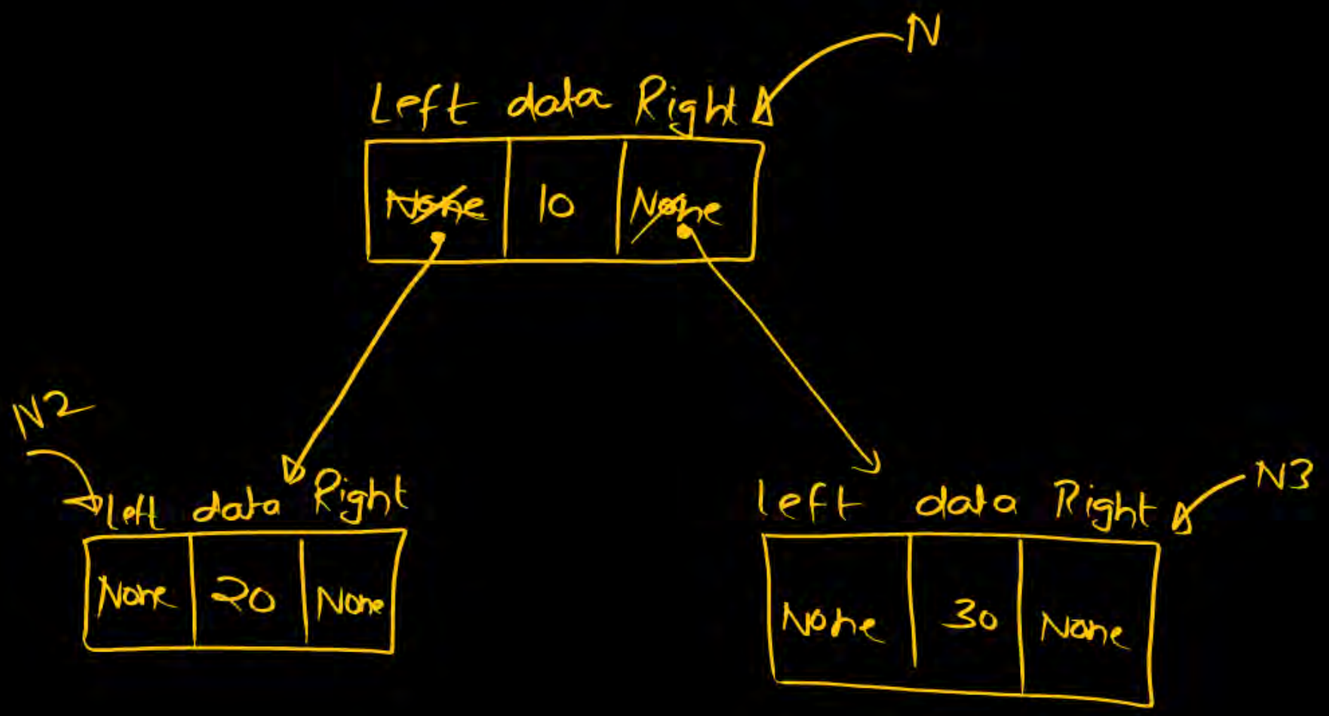




Class BTNode :


```
def __init__(self, data):
    self.data = data
    self.Left = None
    self.Right = Node
```


```
N1 = BTNode(10)
N2 = BTNode(20)
N3 = BTNode(30)
```



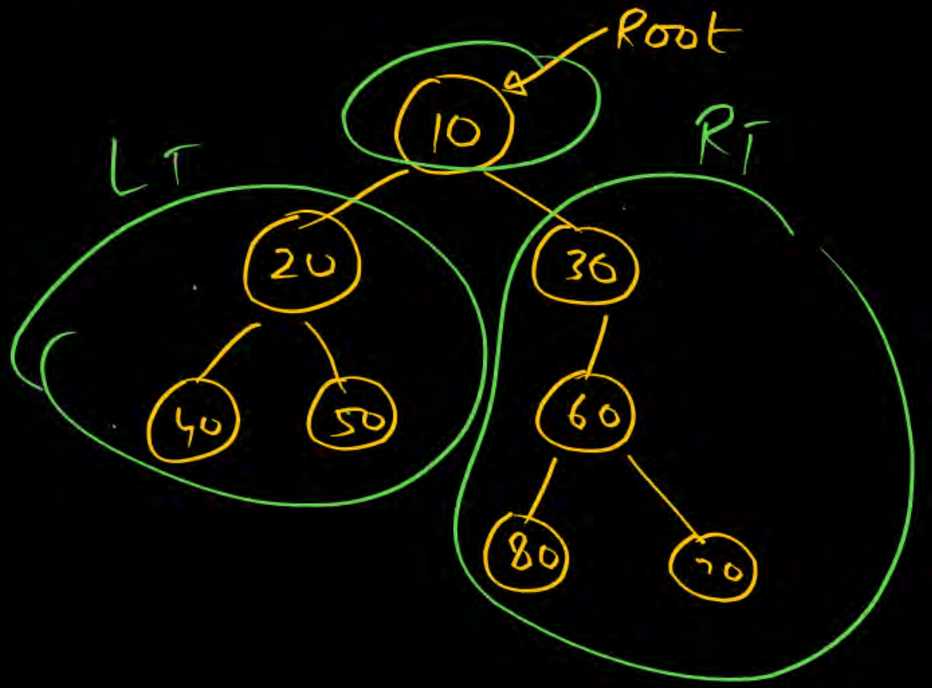
N1.Left = N2

N1.Right = N3

Traversals

Recursive ⇒

10, L_T, R_T



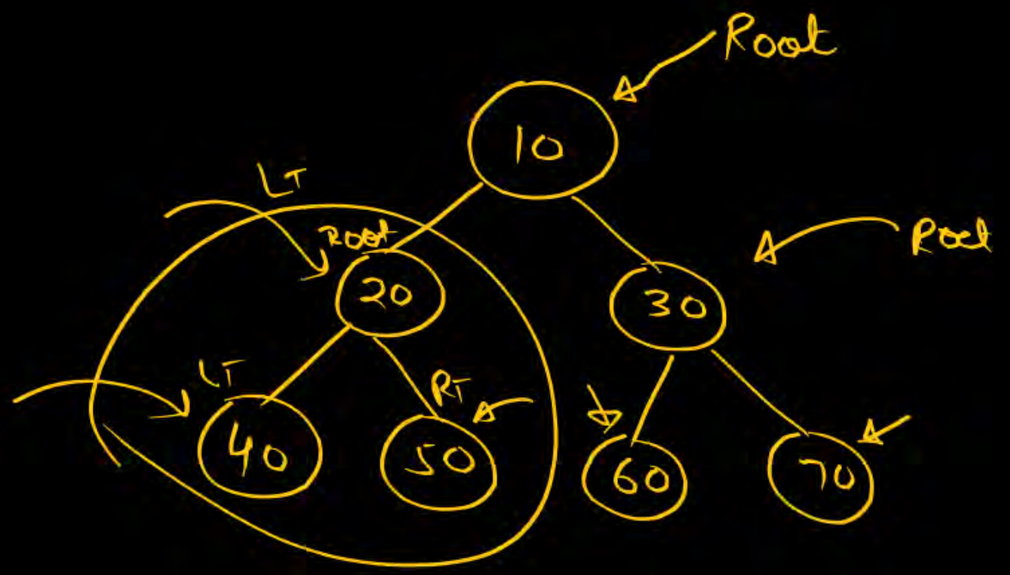
Pre-Order

Rooted Tree

① Print root

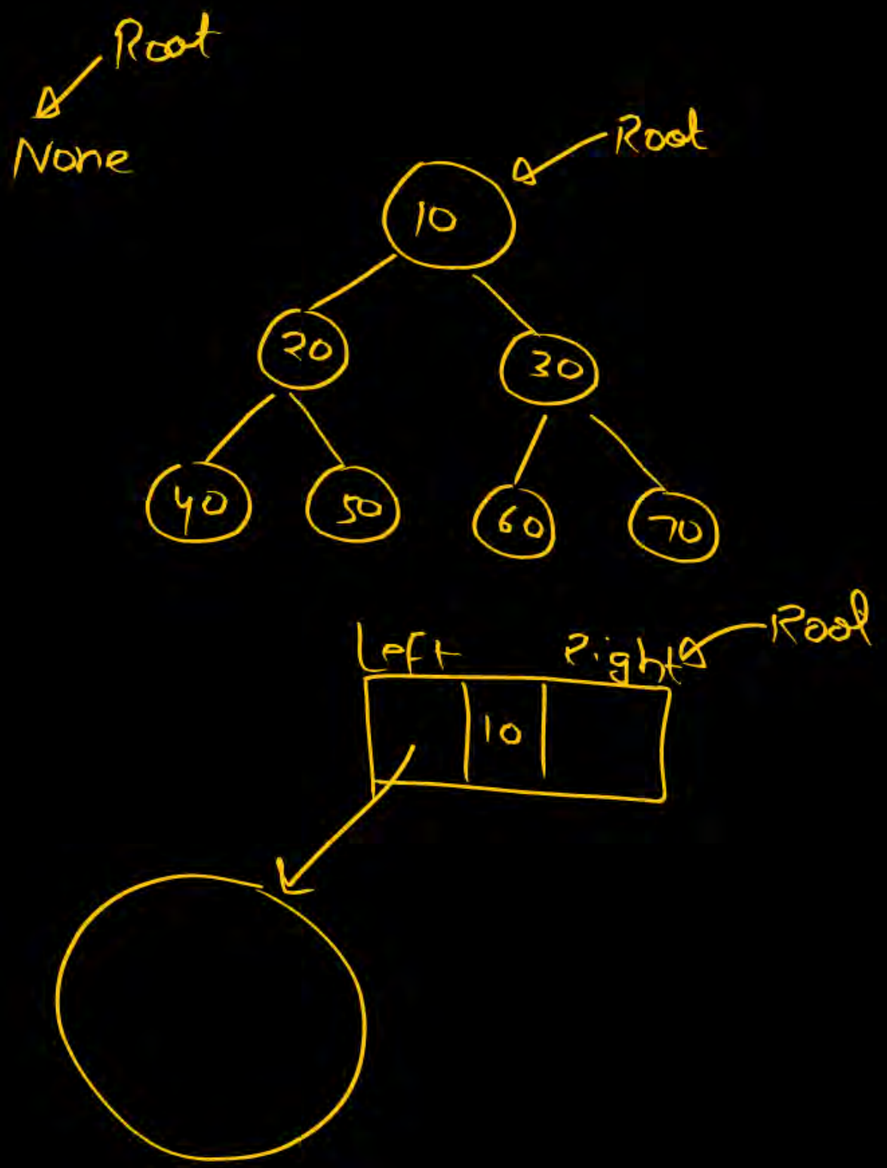
Rec. ② Traverse LT of root in Preorder

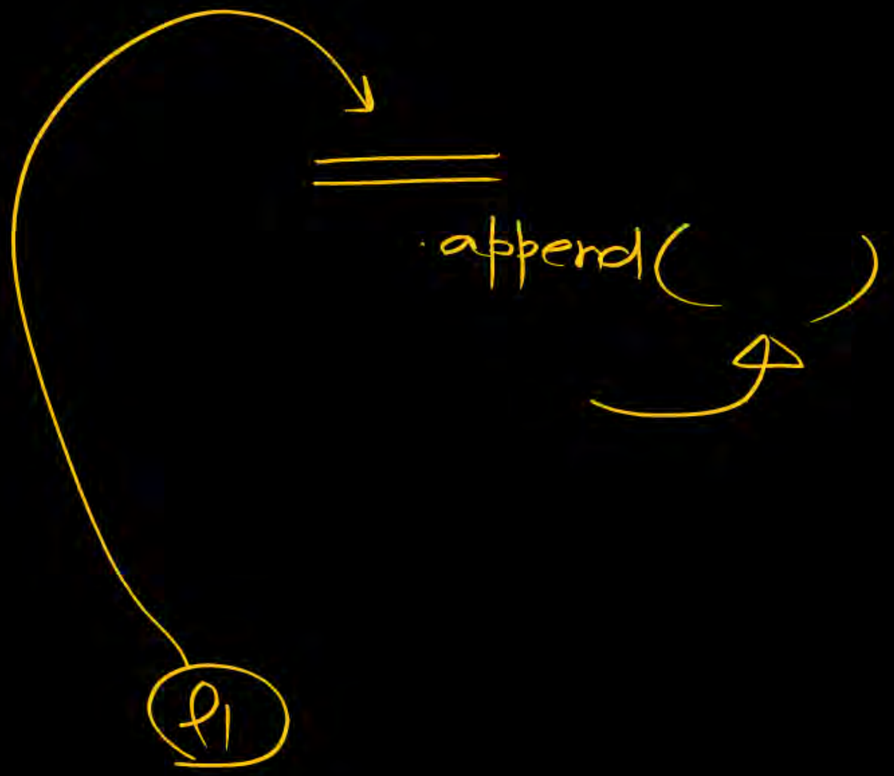
Rec. ③ Traverse RT of root in Preorder



10 20 40 50 30 60 70

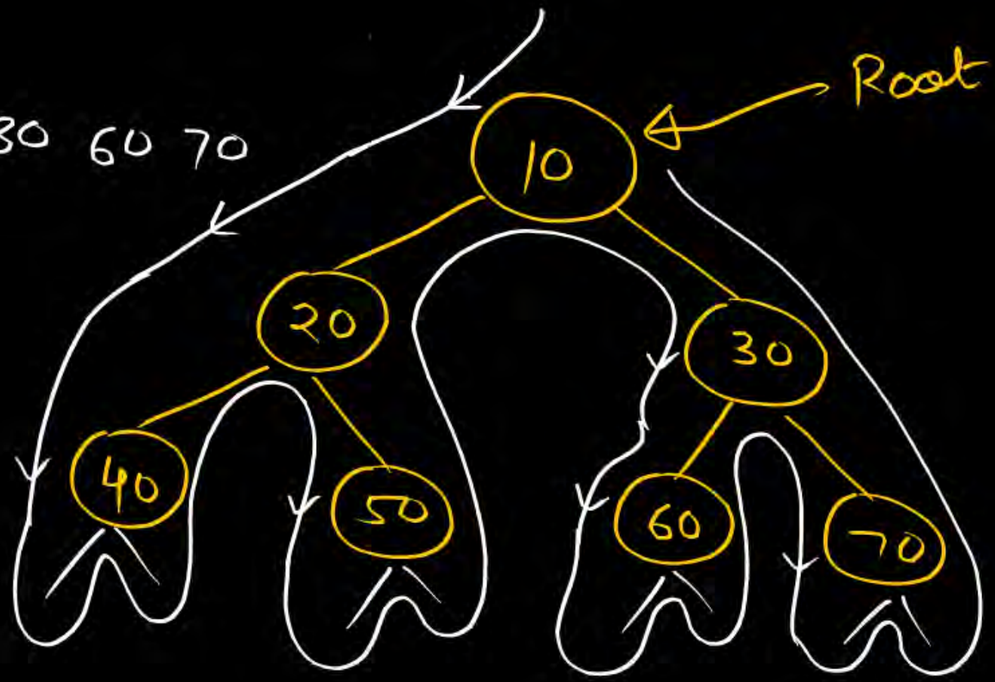
```
def Pre(Root):
    if Root is None:
        return
    print(Root.data)
    Pre(Root.Left)
    Pre(Root.Right)
```




.append( )
q1

Preorder

10 20 40 50 30 60 70



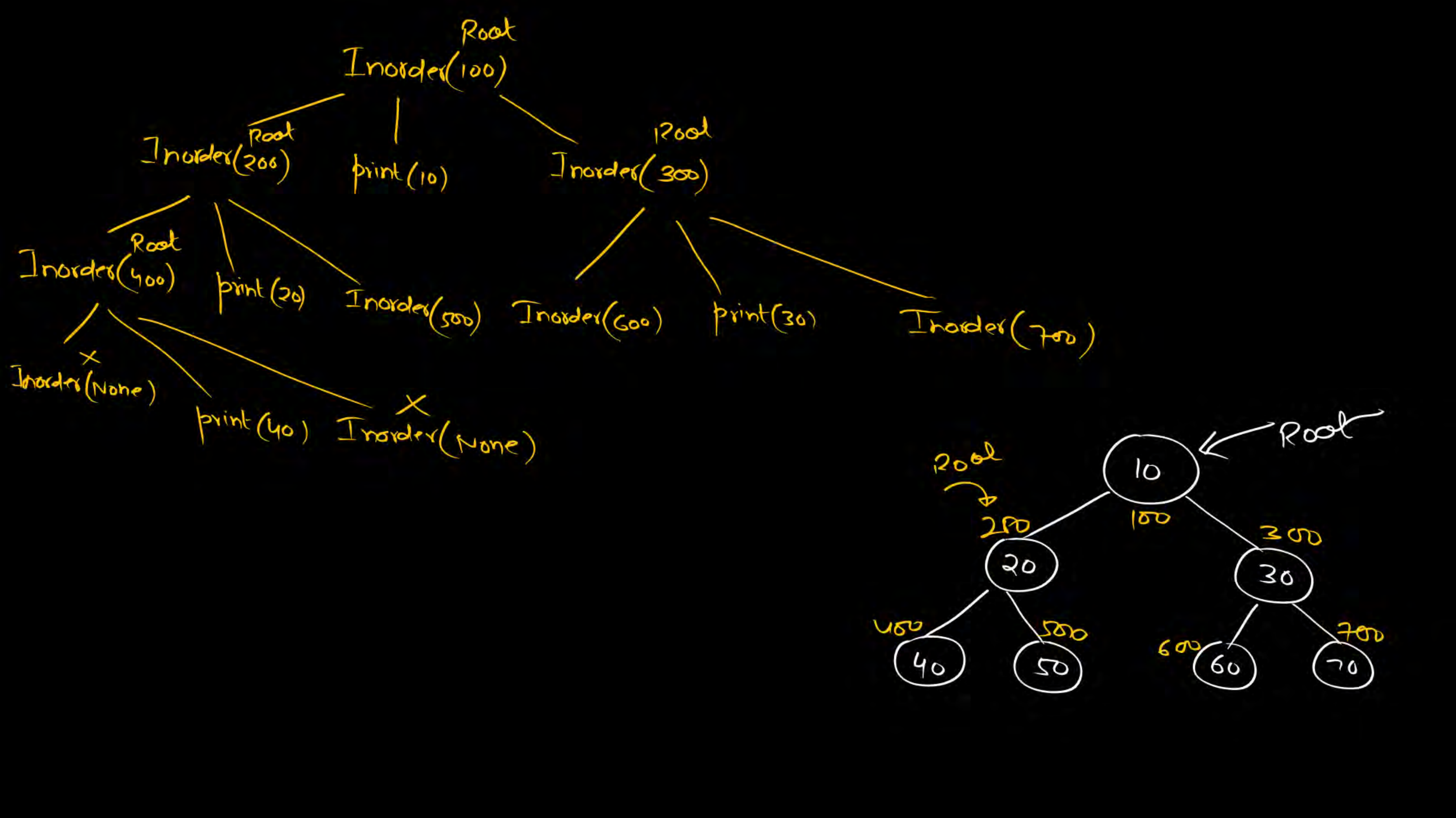
Inorder

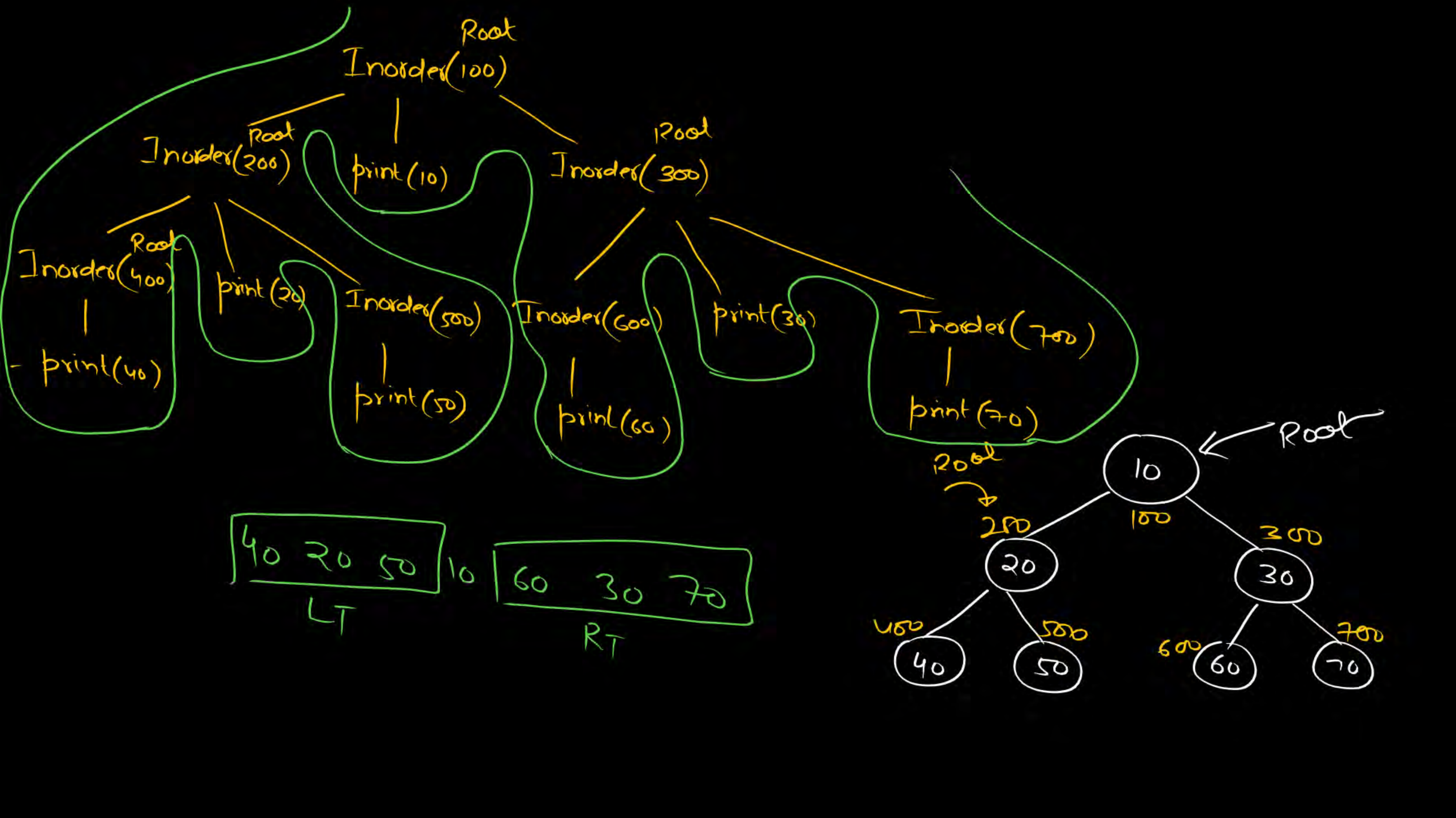
Inorder

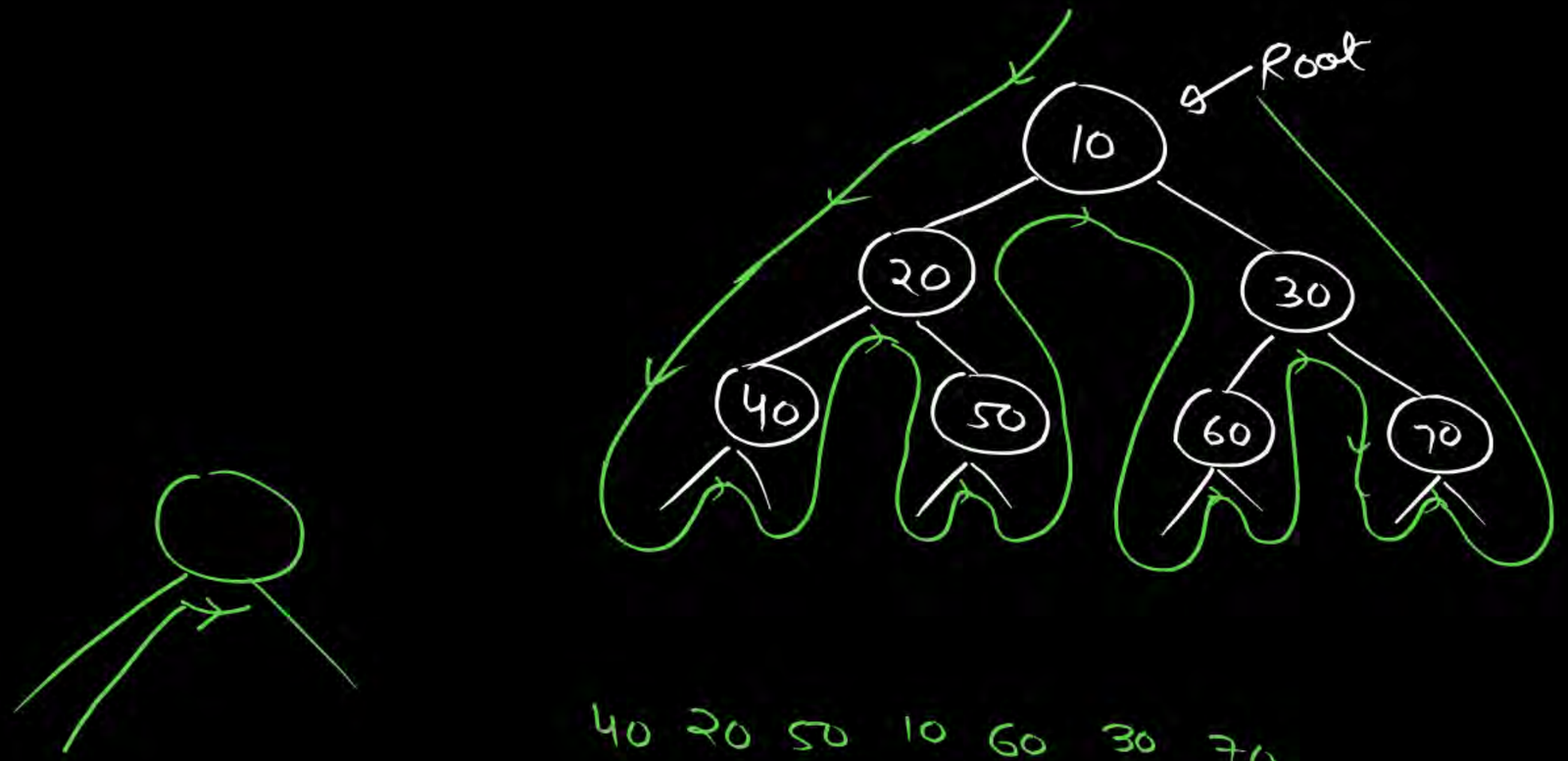
1.) Traverse L_T of root in Inorder

2.) print root

3.) Traverse R_T of root in Inoider


Root
Inorder(100)
Root
Inorder(200)
print(10)
Root
Inorder(300)
Root
Inorder(400)
print(20)
Inorder(500)
Inorder(600)
print(30)
Inorder(700)
Inorder(None)
print(40)
Inorder(None)
Root
200
100
300
10
20
30
400
500
600
700
40
50
60
70

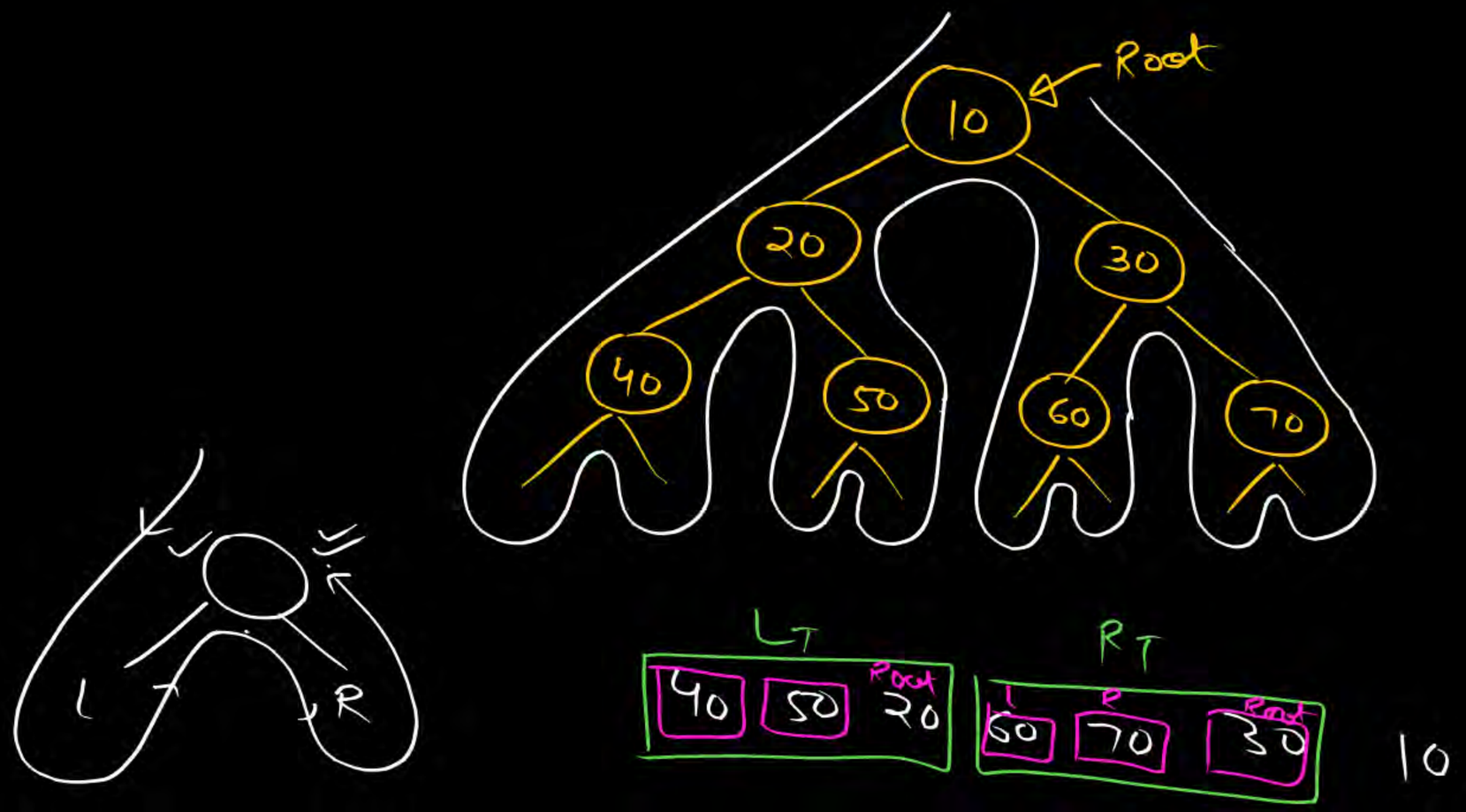

Root
Inorder(100)
Root
Inorder(200)
print(10)
Root
Inorder(300)
Root
Inorder(400)
print(20)
Inorder(500)
Inorder(600)
print(30)
Inorder(700)
print(40)
print(50)
print(60)
print(70)
40 20 50
10
60 30 70
LT
RT
Root
200
10
250
100
300
20
30
400
500
600
700
40
50
60
70



40 20 50 10 60 30 70

Postorder

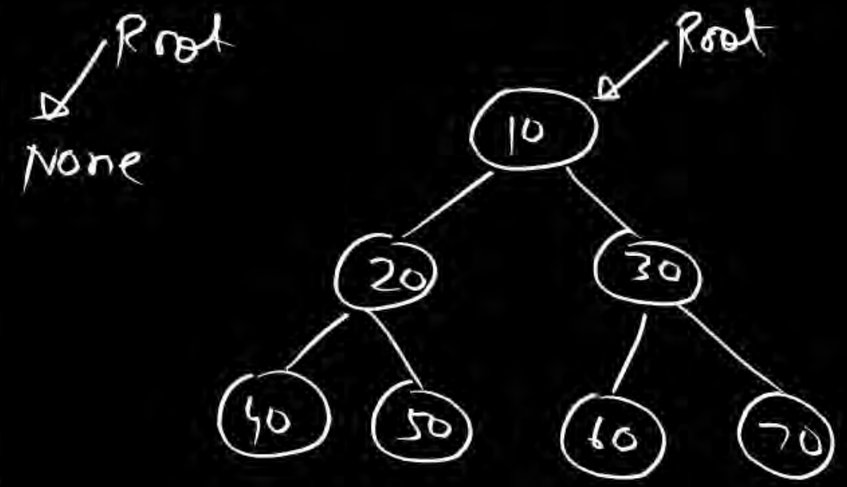
1. Traverse L_T of root in Postorder
2. Traverse R_T of root in Postorder
3. print Root


Root
10
20
30
40
50
60
70
L
R
LT
RT
40
50
Root
20
60
70
Root
30
10

1. Count nodes

HW — 30 minutes

```
c=0
def count(Root):
    if Root is None:
        return 0
```



t.me/PWpankajsirP

In [3]:
```
class BTNode:
    def __init__(self,data):
        self.data=data
        self.Left=None
        self.Right=None
n1=BTNode(10)
n2=BTNode(20)
n3=BTNode(30)
n1.Left=n2
n1.Right=n3
n4=BTNode(40)
n5=BTNode(50)
n6=BTNode(60)
n7=BTNode(70)
n2.Left=n4
n2.Right=n5
n3.Left=n6
n3.Right=n7
```

In [4]:
```
def Pre(root):
    if root is None :
        return
    print(root.data)
    Pre(root.Left)
    Pre(root.Right)
Pre(n1)
```

```
10
20
40
50
30
60
70
```

In [7]:
```
def Inorder(root):
    if root is None:
        return
    Inorder(root.Left)
    print(root.data)
    Inorder(root.Right)
```

In [8]:
```
Inorder(n1)
```

```
40
20
50
10
60
30
70
```

In [9]:
```
def Postorder(root):
    if root is None:
        return
    Postorder(root.Left)
    Postorder(root.Right)
```

```
    print(root.data)
```

In [10]:
```
Postorder(n1)
```

```
40
50
20
60
70
30
10
```

In [ ]:
```

```

THANK - YOU