IX. Binary tree implementation

TREE

→ Binary → 0
→ 1

Root
0 1 2   (Not more than 3 child)
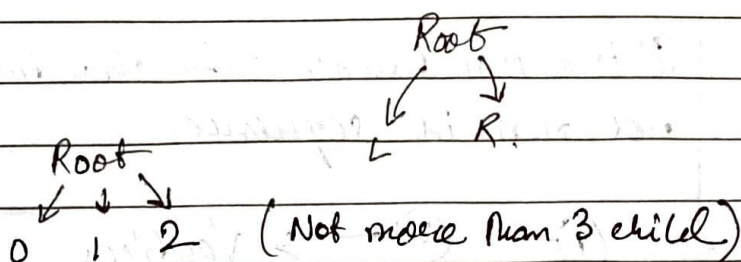
Root
L   R.

```
Root Node

Left Branch    Right Branch
```

Node

LEFT	NODE	RIGHT

```python
class Node:
    def __init__(self, data) -> None:
        self.left = None
        self.right = Node
        self.data = data
```

Syntax to print left & Right Node

```python
    def show(self):
        print(self.data)

root = Node(100)
root.show()

root.left = 99
root.right = 101
root.show()
```

↑ This is not applicable we need to create the object Lets, do it.

```python
    def show(self):
        if self.left:
            self.left.show()
        print(self.left data)
        if self.right:
            self.right.show()
        print(self.data)
```

```python
root = Node(100)
r_left = Node(99)      # we have created the object.
r_right = Node(101)    # Remind it.
root.left = r_left
root.right = r_right
root.show()
```