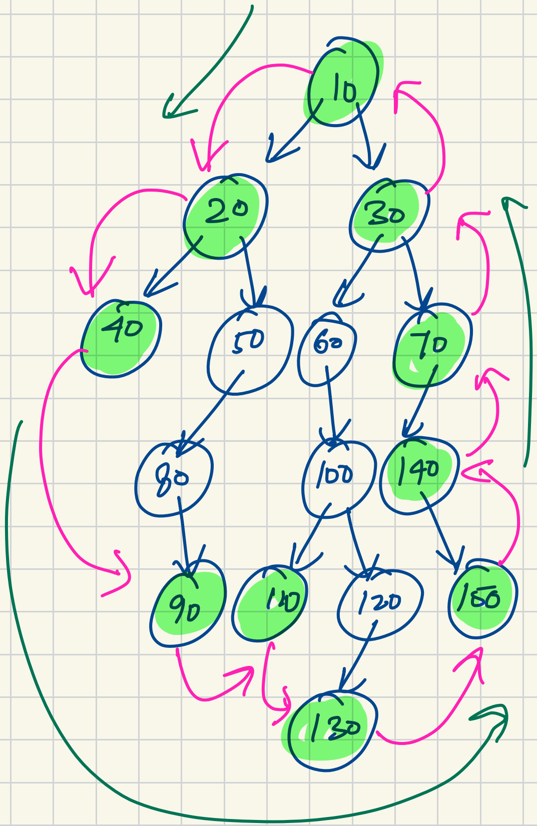
Boundry Traversal



Boundry of the Binary Tree

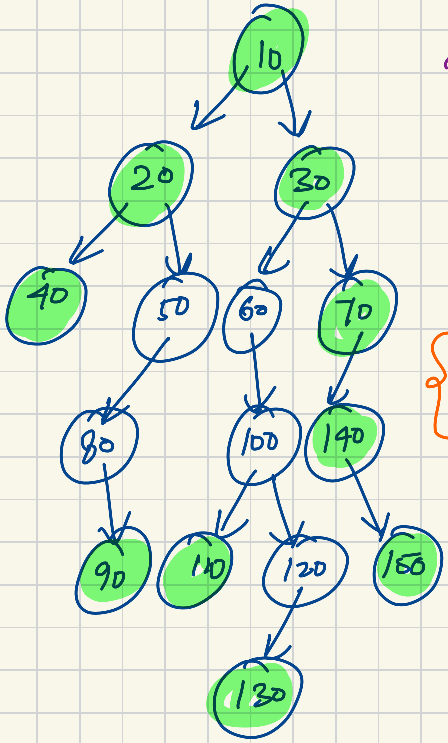
o/p 10, 20, 40, 90, 110, 130, 150, 140, 70, 30

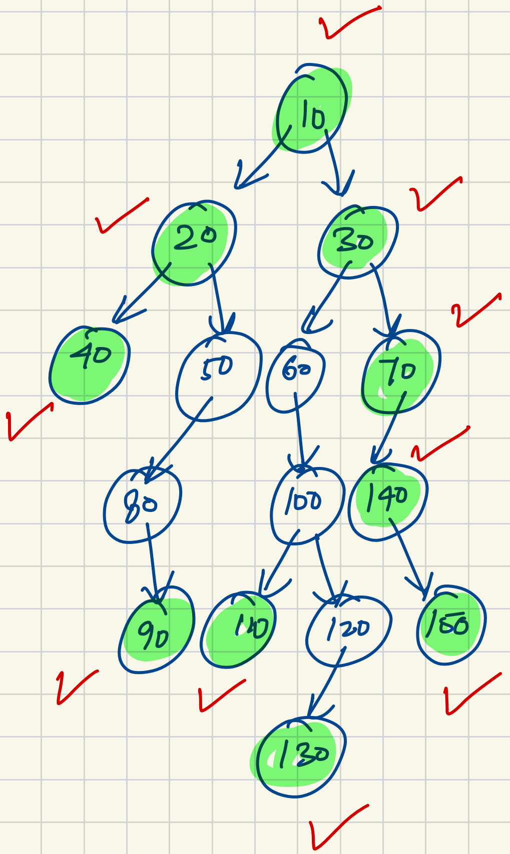
Boundary Traversal:

↗ left Boundary + leafNodes + Right-Boundary

$\boxed{\text{root} + LB + LN + RB}$

{make sure you ignore leaf Nodes}





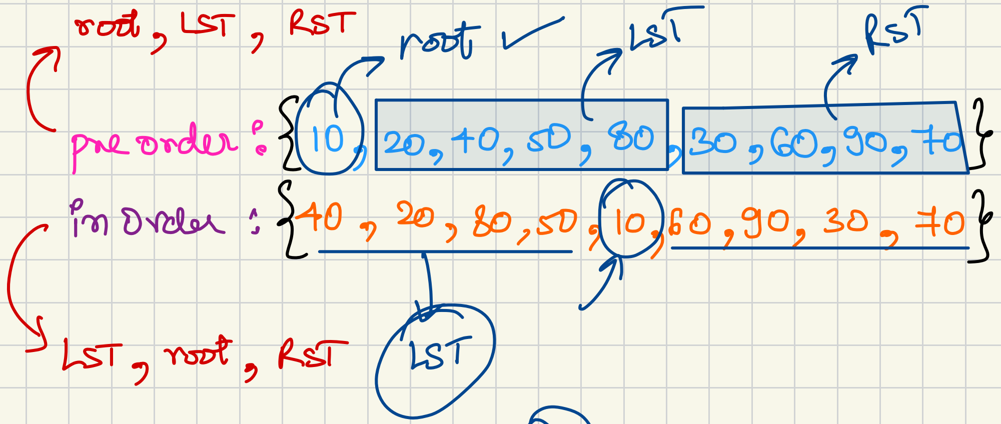
for LN:

```
{ if (node.left == null &&
                node.right == null)
        → you are a leaf Node.
}
```

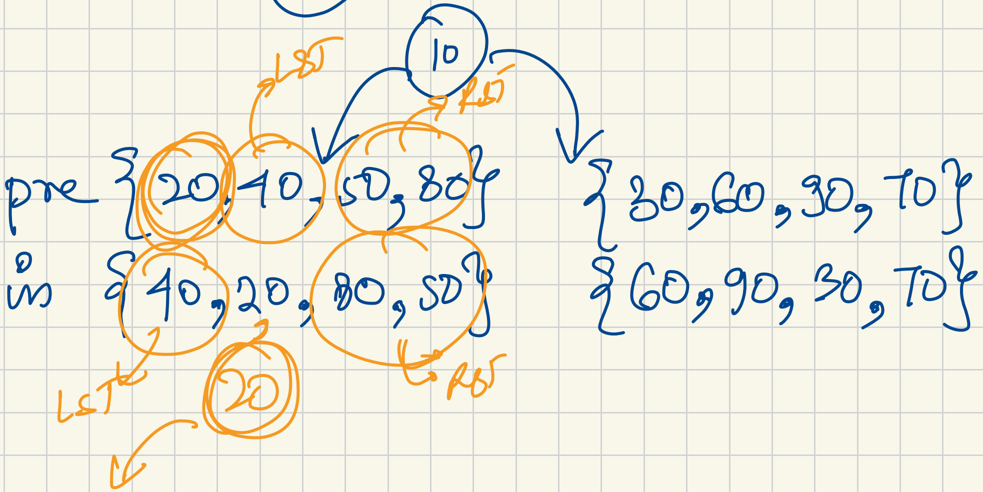
for LB:

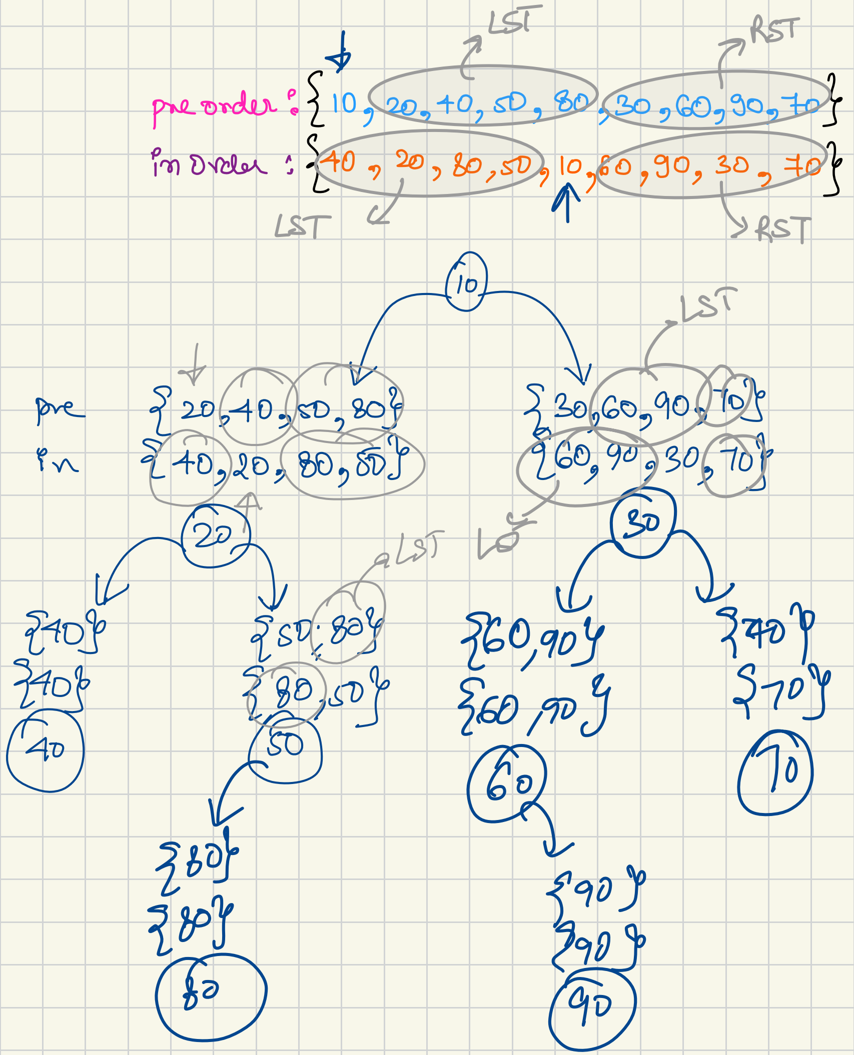
```
if (Node.left != null)
        go left
else
        go right
```

Build a Binary Tree using, pre order and in order traversal.



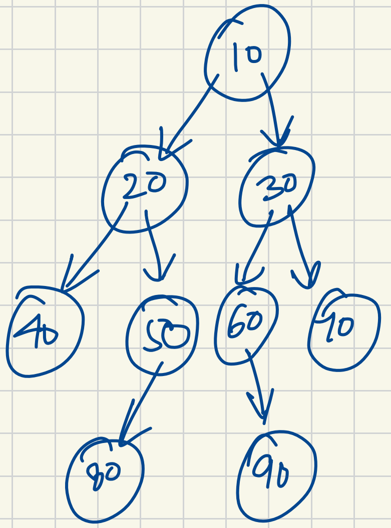
root, LST, RST
root ✓
LST
RST
pre order : {10, 20, 40, 50, 80, 30, 60, 90, 70}
in Order : {40, 20, 80, 50, 10, 60, 90, 30, 70}
LST
LST, root, RST

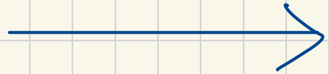


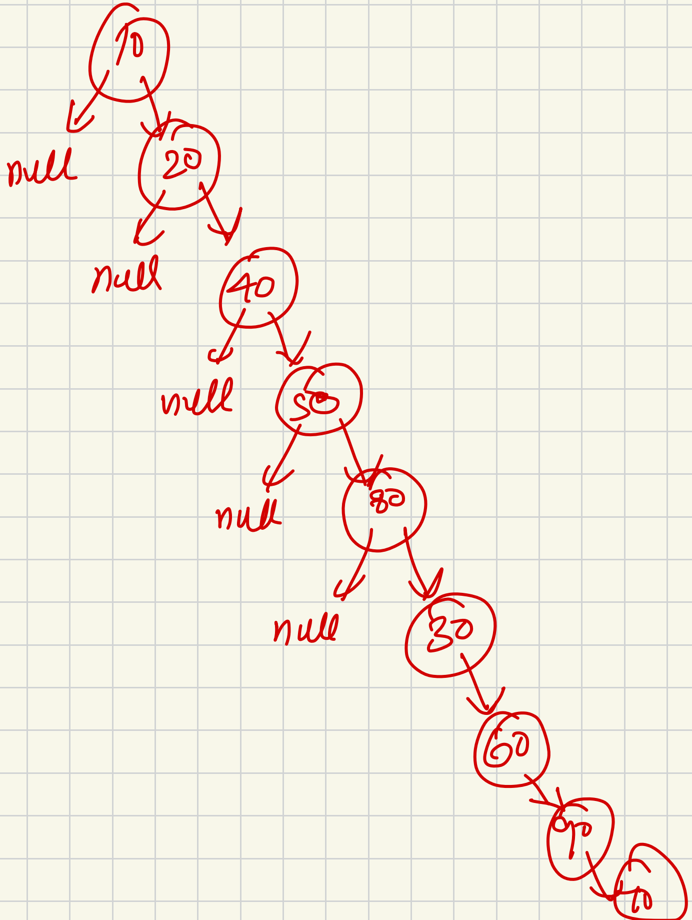


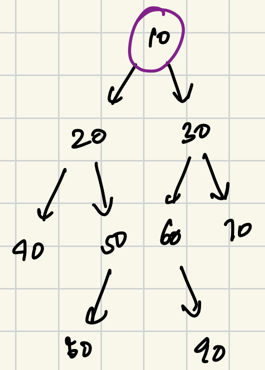
pre order : {10, 20, 40, 50, 80, 30, 60, 90, 70}
in order : {40, 20, 80, 50, 10, 60, 90, 30, 70}
LST
RST
LST
RST
10
pre {20, 40, 50, 80}
in {40, 20, 80, 50}
{30, 60, 90, 70}
{60, 90, 30, 70}
LST
LST
LST
20
30
{40}
{40}
40
{50, 80}
{80, 50}
50
{80}
{80}
80
{60, 90}
{60, 90}
60
{90}
{90}
90
{70}
{70}
70

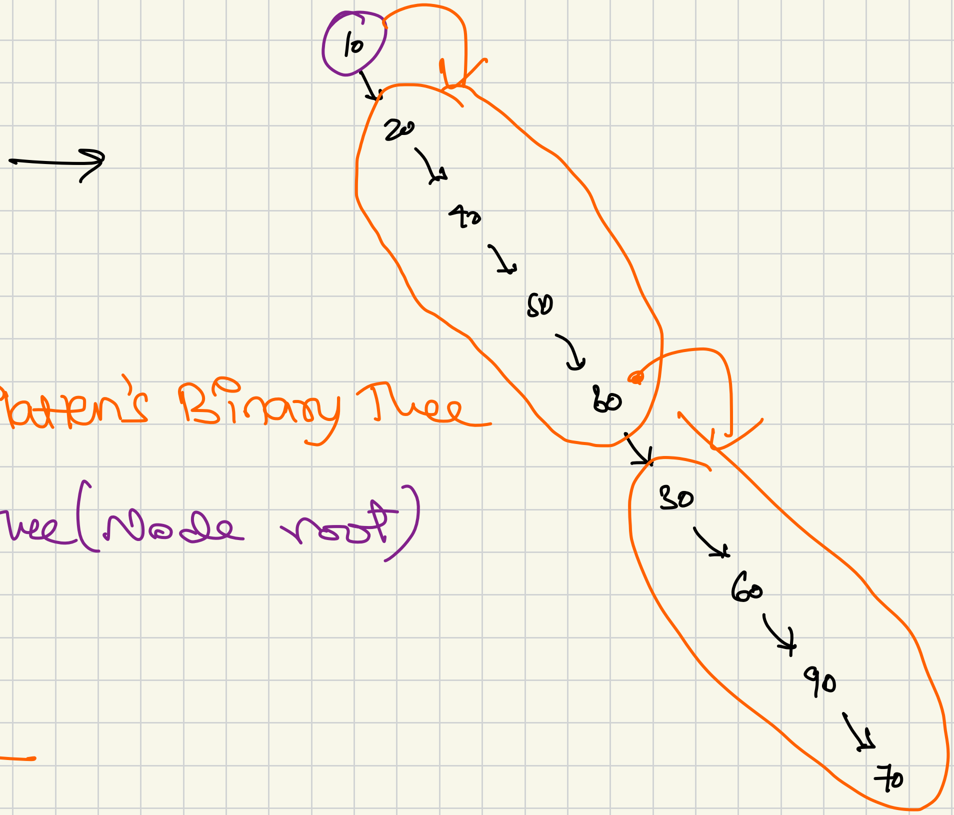
Flatten a Binary Tree









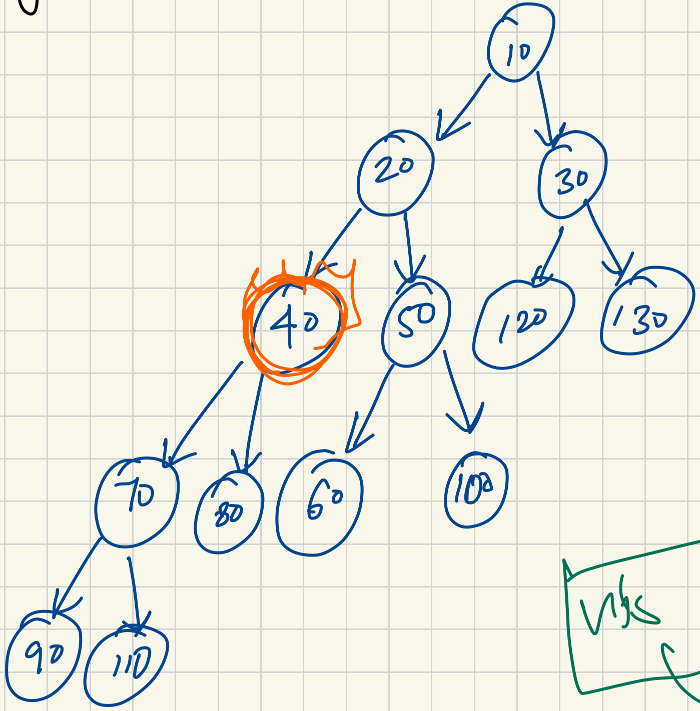


faith: flatten's Binary Tree

void flattenBinaryTree(Node root)
{

Burning of BT

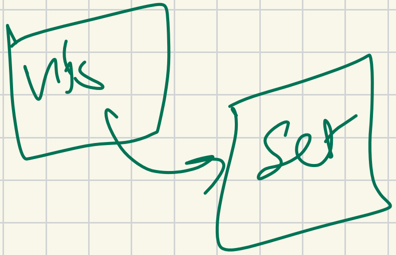
→ minimum time to burn entire tree }



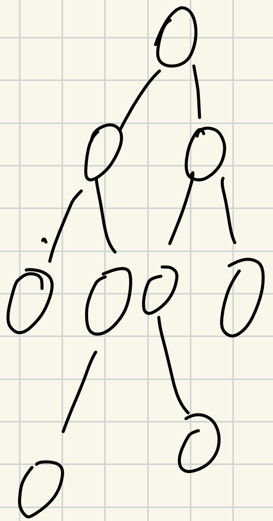
TC: O(1)

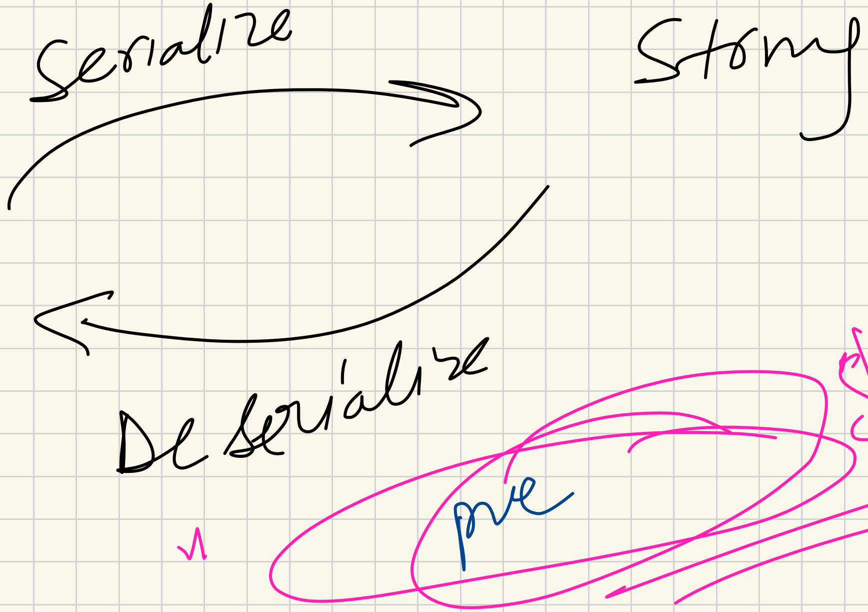
HashMap

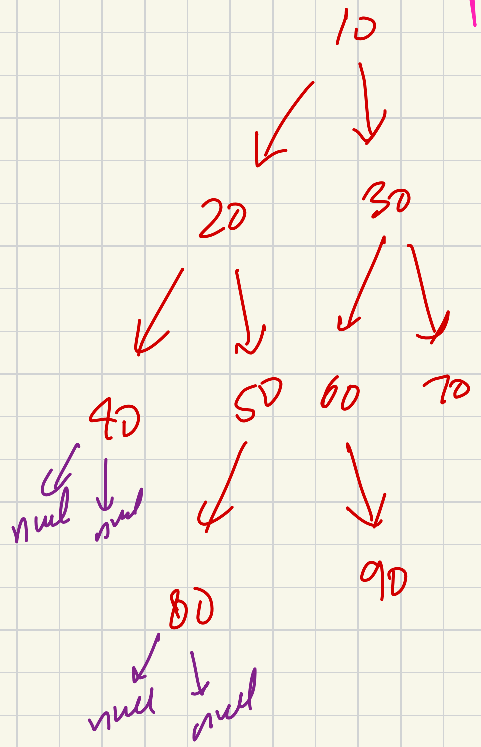
{child → parent}



Serialize & De serialize a BT

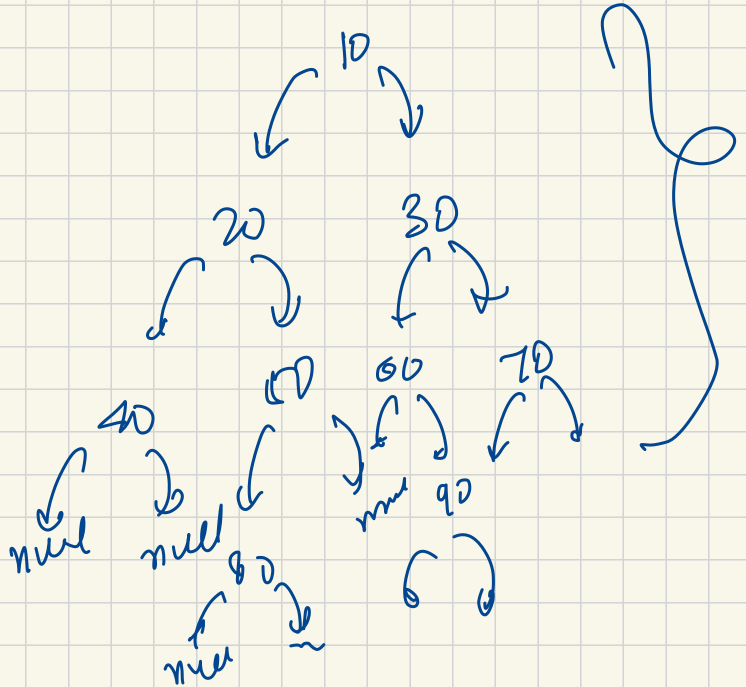






Pre

10, 20, 40, null, null, 50, 80, null, null, null, 30, 60, null, 90,
null, null, 70, null, null.


10
20
30
40
50
60
70
80
90
null
null
null
null