Agenda

- Size, Sum, Max^{m} and height
- Balanced Binary Tree
- Diameter of Binary Tree
- Level order Traversal
- Left view
- Zig zag Traversal

Size of Binary Tree

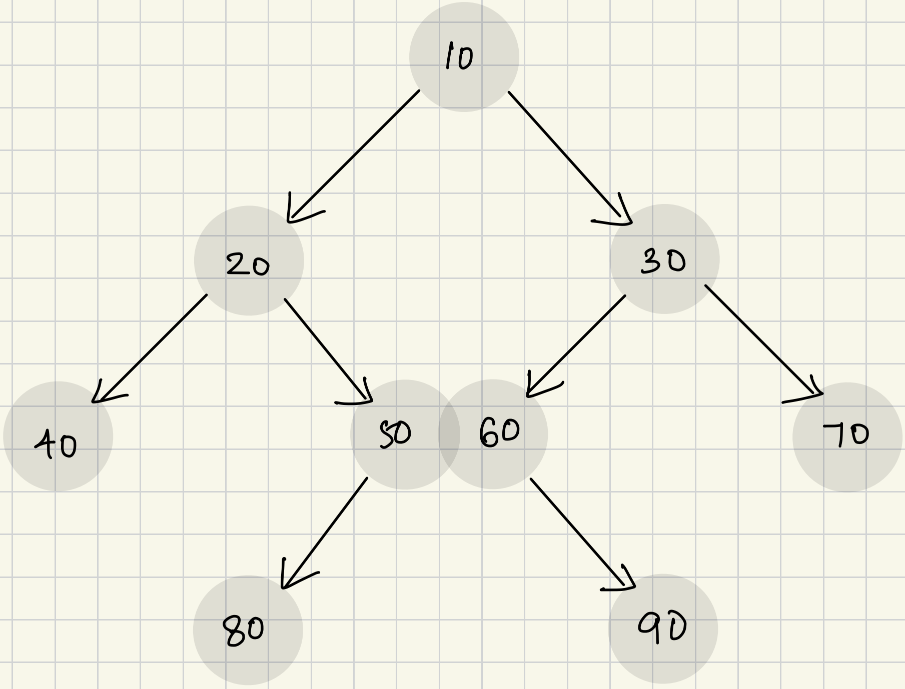
: No. of Nodes in the given BT.

size = 9 Nodes

why?

faith:

(n-1) result



~~int fun(Node root)~~

~~{~~

~~if (root = = null)~~

~~return 0;~~

~~int a = fun(root.left)+1;~~

~~int b = fun(root right)+1;~~

~~return a+b;~~

~~}~~

$\boxed{\text{ToH}}$

↓

though?

printdec (int n)

↓

Question

$n \downarrow 0$

$fun(n) = n + fun(n-1)$

feith

~~#~~ { print no. from n → 0 in dec

Base Case

$\boxed{n \geq 0}$

$fun(n) = n, [(n-1), (n-2), \ldots\ldots, 3, 2, 1]$

$fun(n-1) = (n-1), (n-2) \ldots\ldots\ldots 3, 2, 1$

faith

```
fun(  )
{
    Base Case

    recursive call → Expectation
}
```

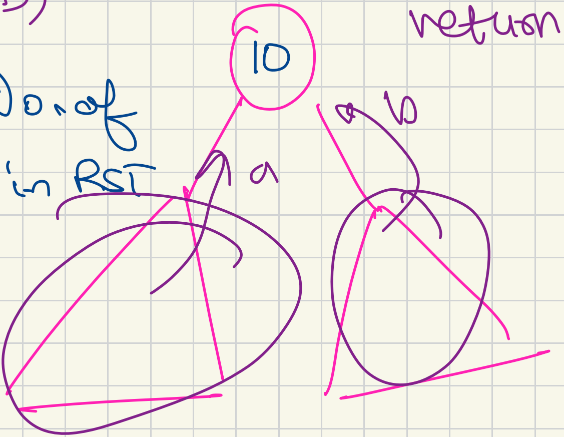
funcn: return size of given Binary Tree from root

```
int fun(Node root)
{
    int a = fun(root.LST);
    int b = fun(root.RST);
    return a+1+b;
}
```

Base Case

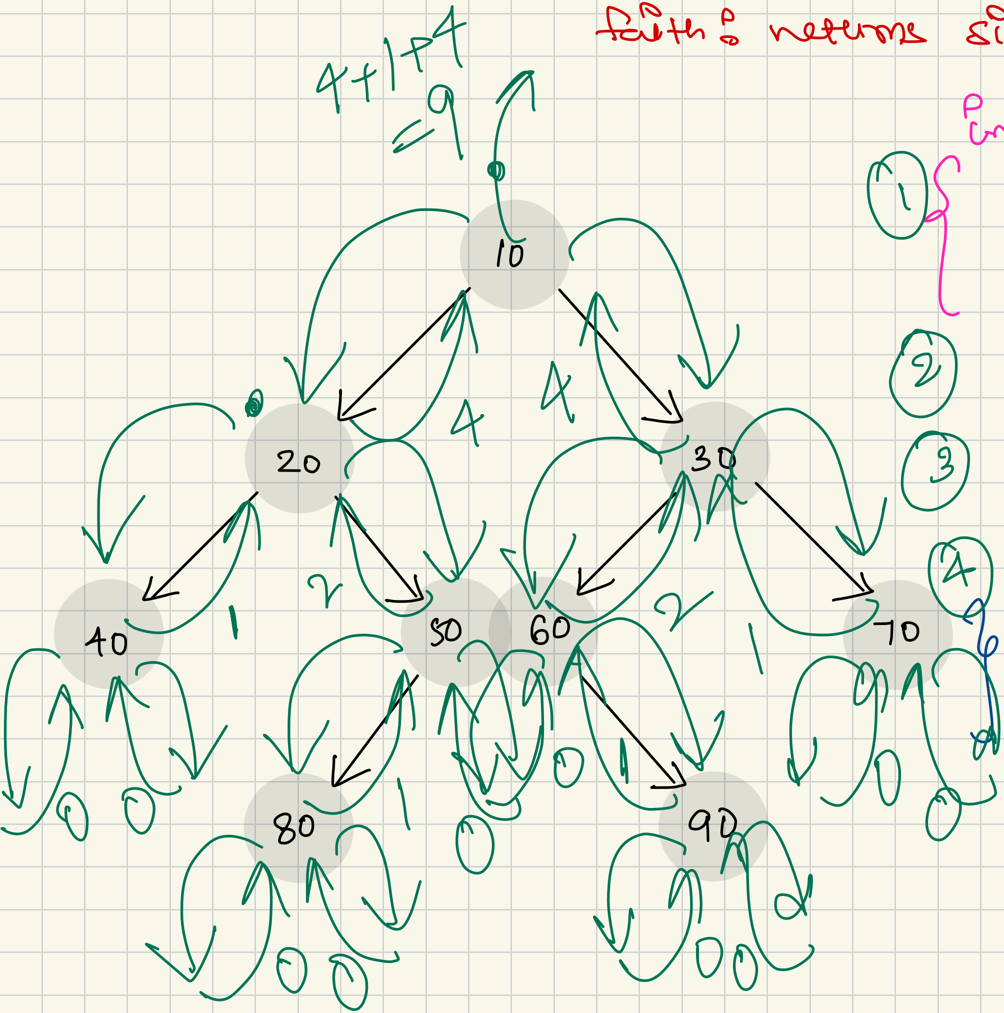
```
if(root==null)
    return 0;
```

fun(10) = No. of Nodes in LST + 1 + No. of Nodes in RST



fun(root) = fun(LST) + 1 + fun(RST)

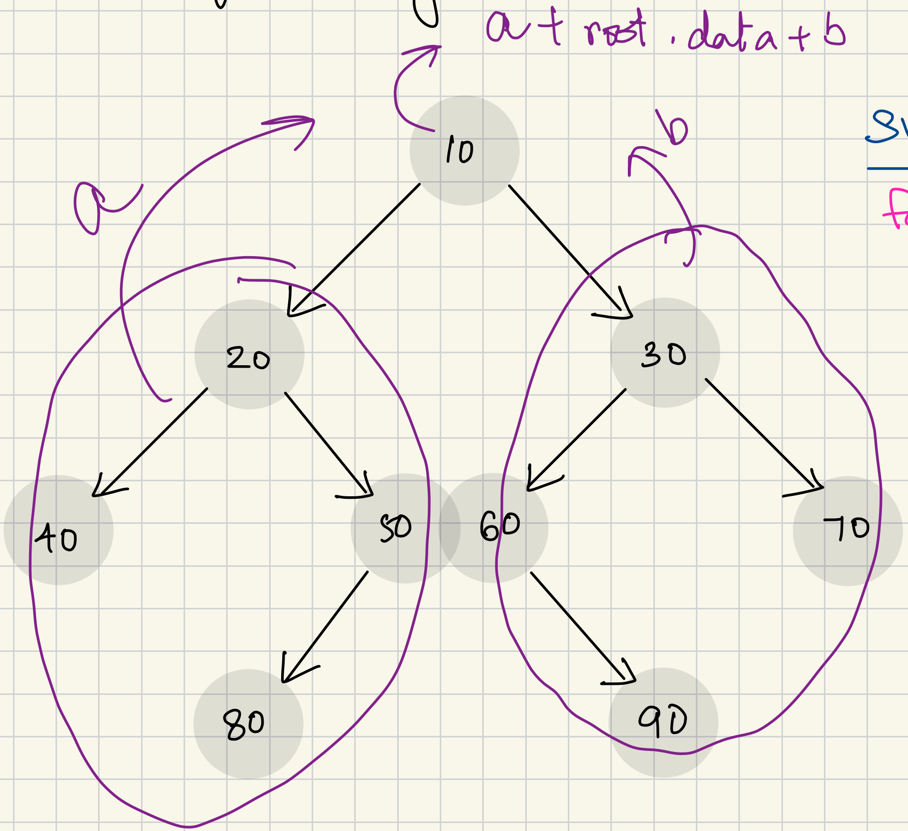
faith: returns size of BT from root



```
int size(Node root)
1 { if(root == null)
        return 0;

2   int a = size(root.left);

3   int b = size(root.right);

4   return a+1+b;
```

Sum of Binary Tree

: sum of all the Nodes of BT

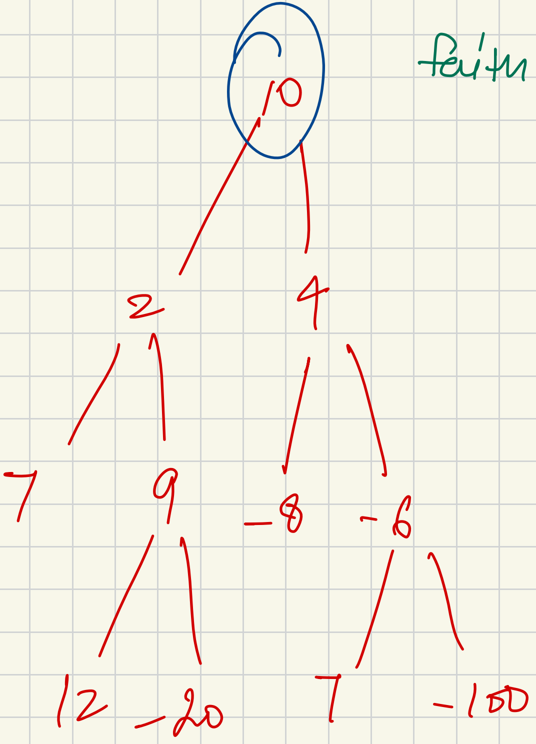


Sum = 450

faith: Sum of BT from given root.

```
int sum(Node root)
{
    if(root == null) return 0;

    int a = sum(root.left);
    int b = sum(root.right);

    return a + root.data + b;
}
```

Maximum of Binary Tree



max^m = 12

faith: max of BT from root

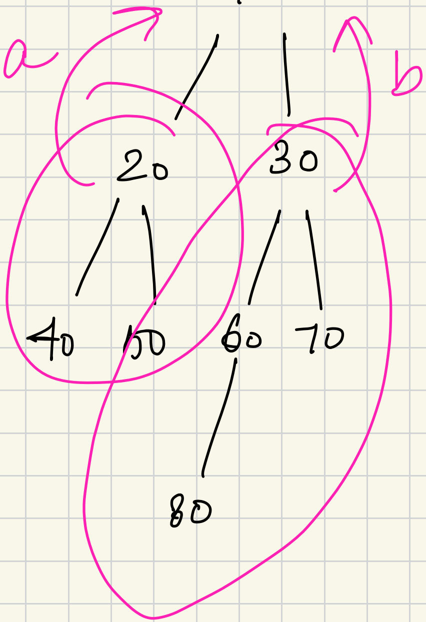
```
int fun( Node root)
{
        if (root == null)
                return -∞;

        int a = fun(root.left);

        int b = fun(root.right);

        return max(a, b, root.data);
}
```

height of given BT

height = 4

faith: 'return h of BT from root'

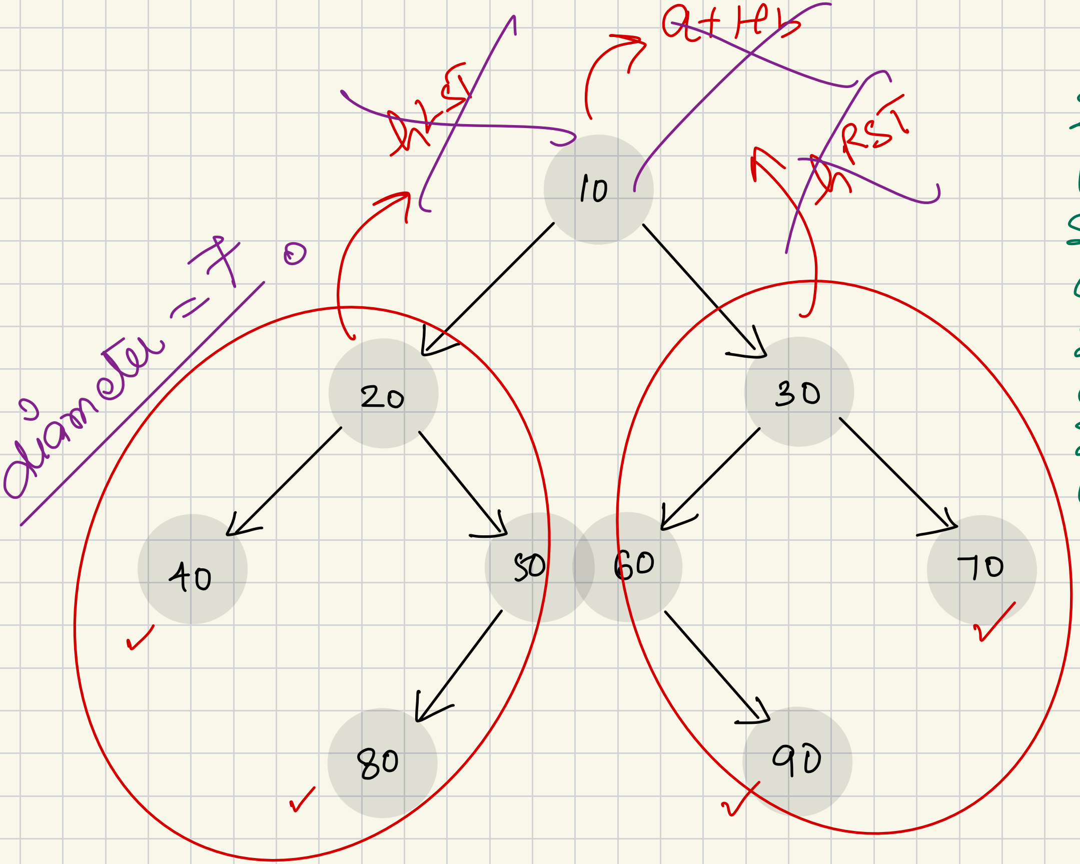
$h = \max(a, b) + 1$



```
int height(Node root)
{

}
```

Diameter of Binary Tree

{Max^{m} dist. b/w any two leaf Nodes}



{40, 80} → 4

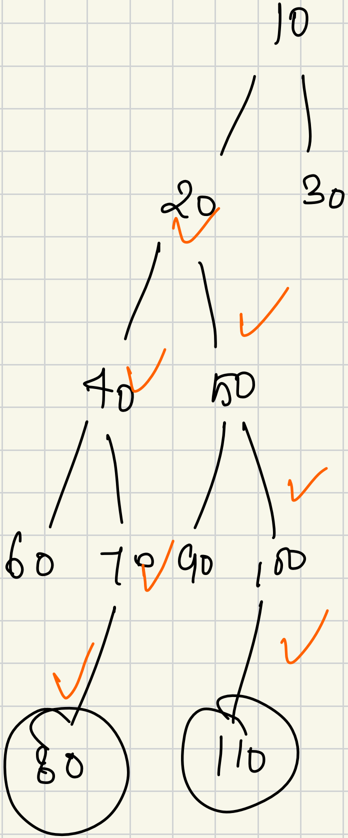
{40, 90} → 6

{40, 70} → 5

{80, 90} → 7

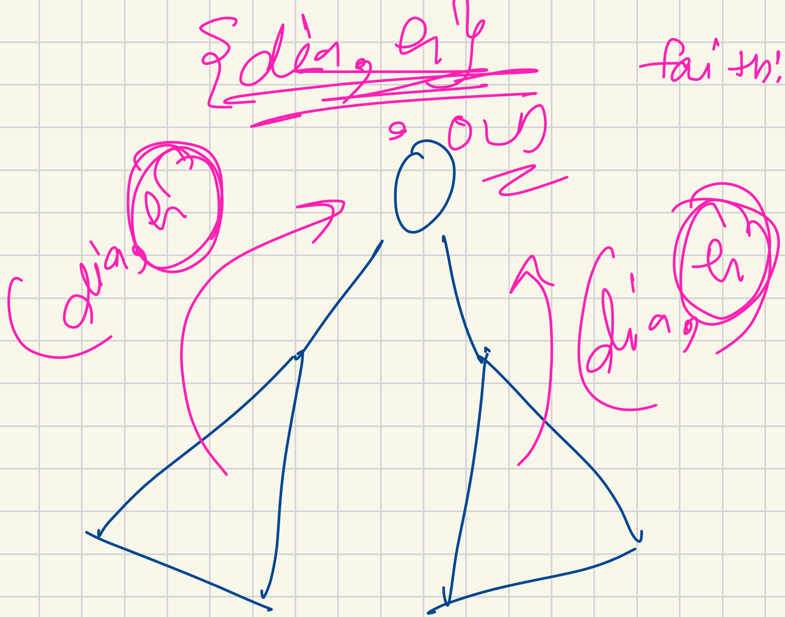
{80, 70} → 6

{90, 70} → 4



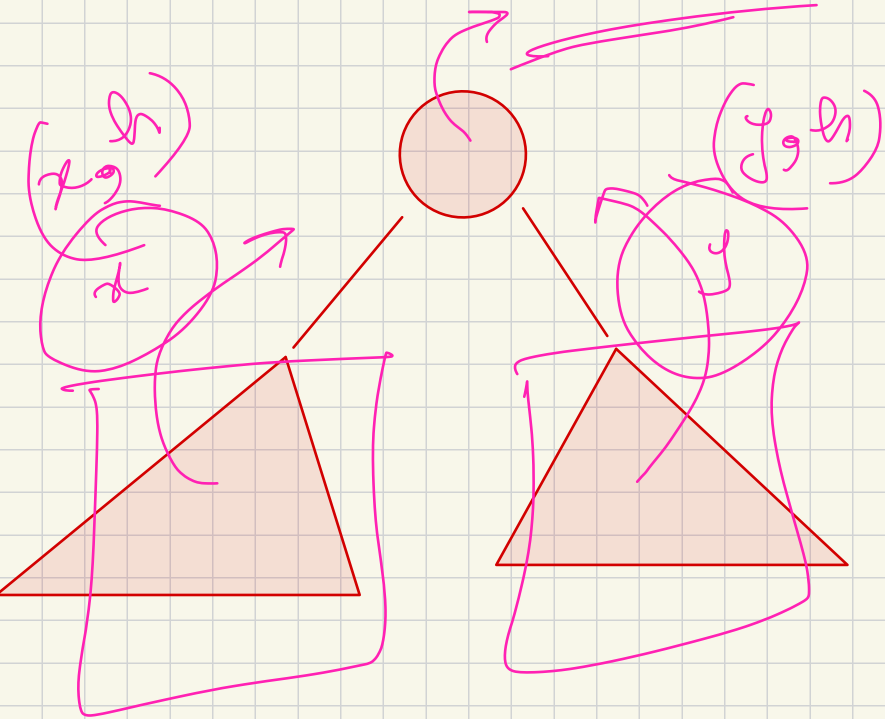
$\boxed{6}$ ✓

NOTE : Dia. is not ment to pass through root



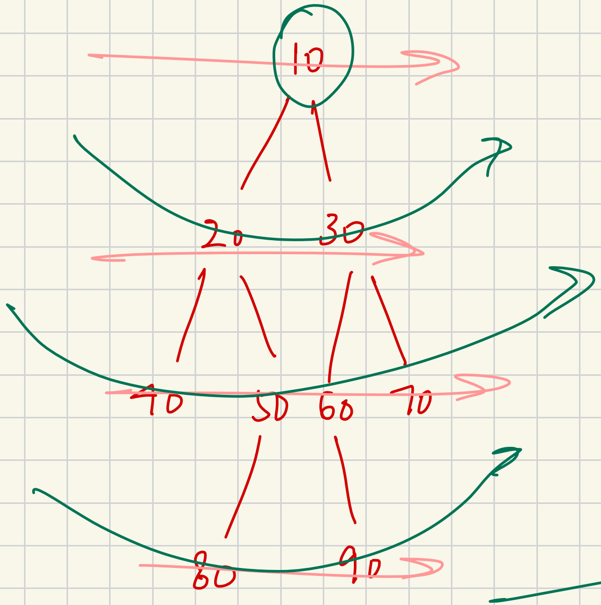
faith: {dia, h} of the tree

On Balanced Binary Tree



Level Order Traversal

BFS

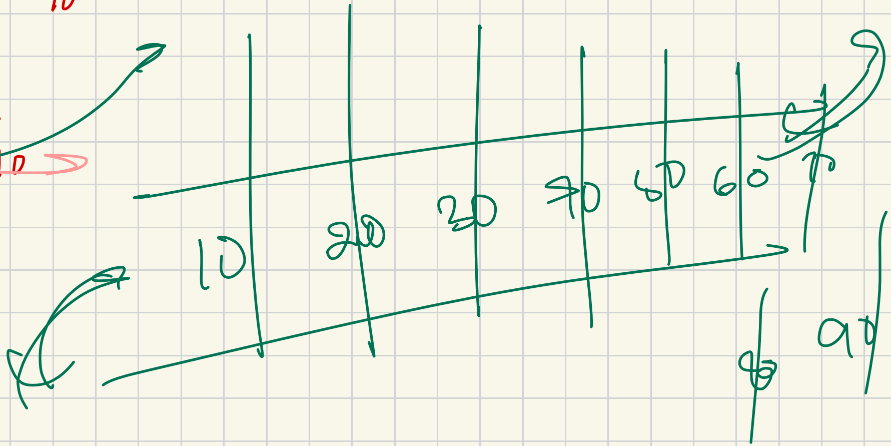


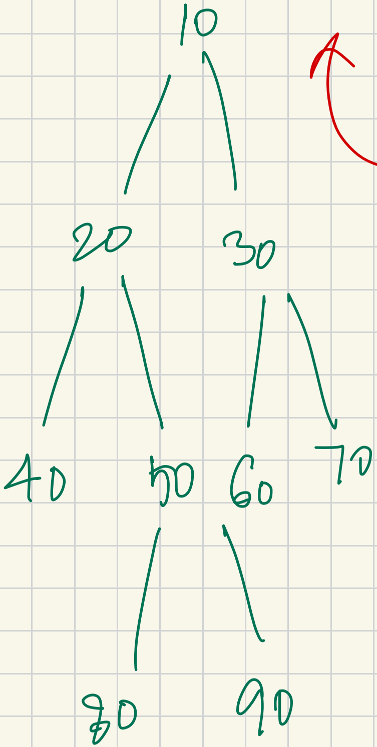
10

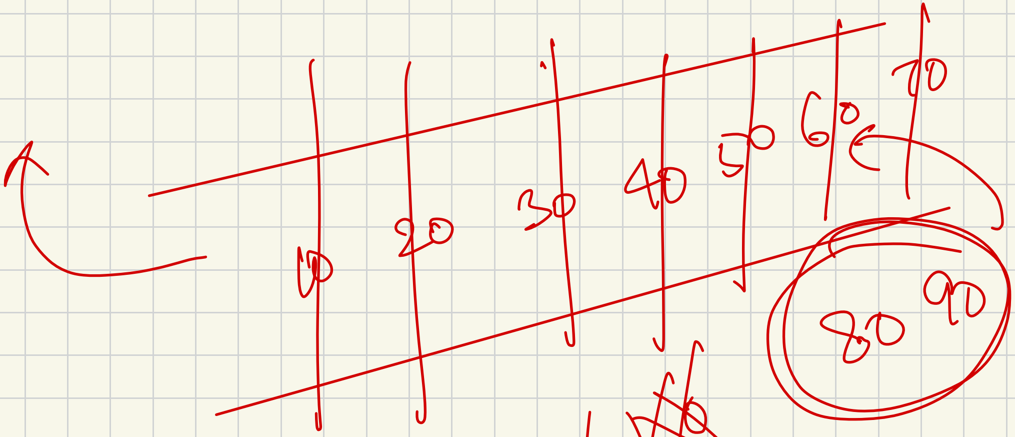
20 30

40 50 60 70

80 90

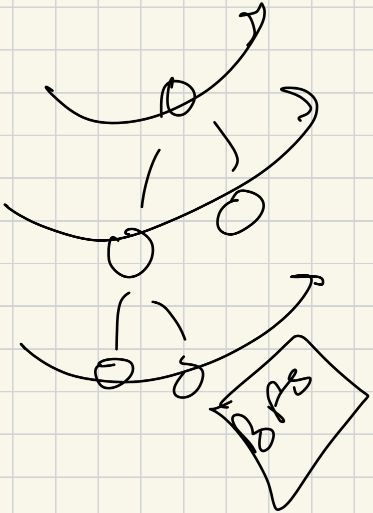


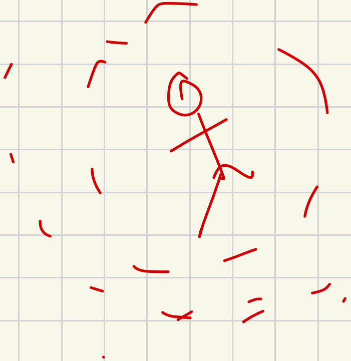


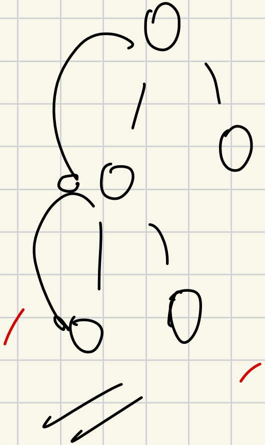


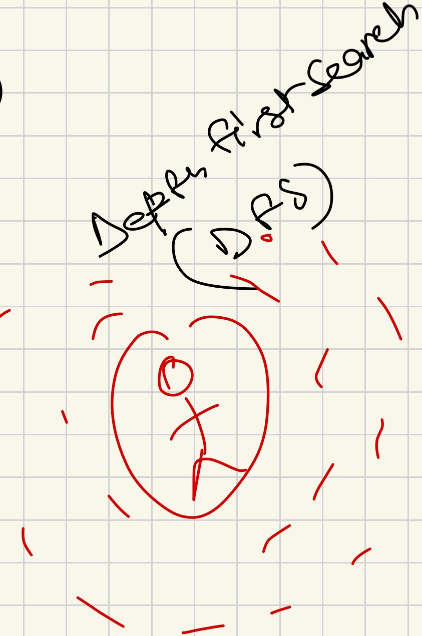
Size = ~~1~~ ~~0~~ ~~2~~ ~~1~~ ~~0~~ ~~4~~ ~~3~~ ~~2~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~

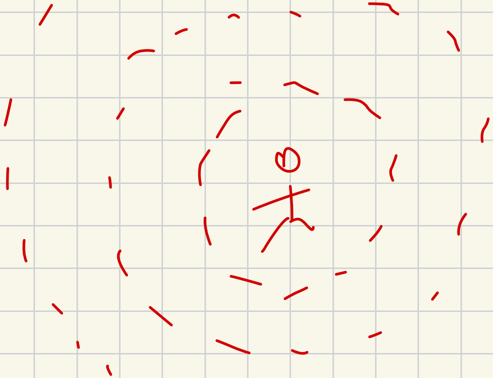
2 ✓


BFS



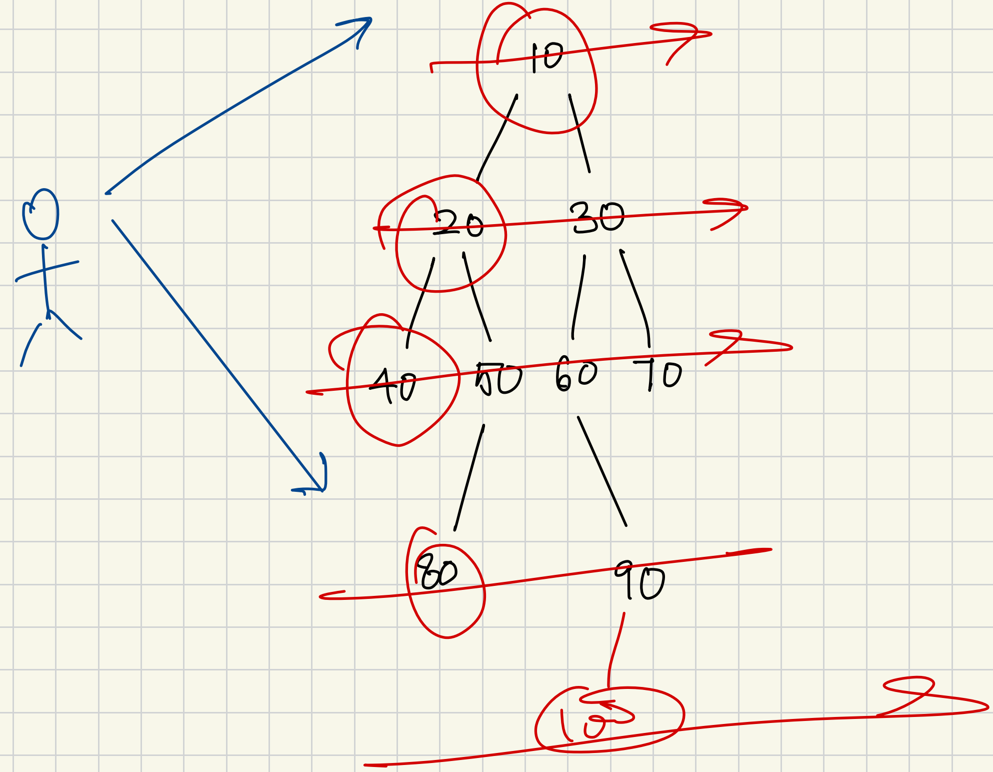



Depth first search
(DFS)



Left View in a Binary Tree.

LV: {10, 20, 40, 80}


10
20
30
40
50
60
70
80
90
100