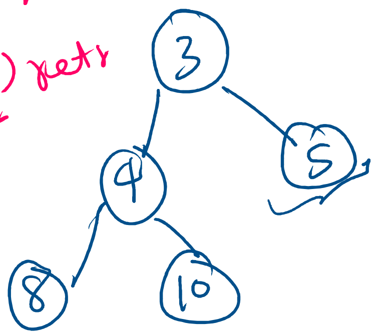
3.18 - Binary Trees 3

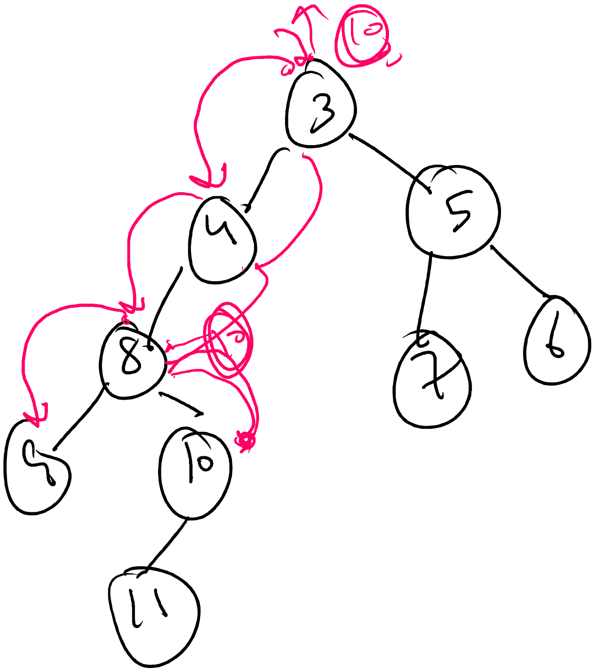
Sunday, August 17, 2025 10:36 AM

```
Node findNode ( Node root, int target) {
    if( root.data == target) ret root;
    Node left= findNode (root.left, t);
    if(left != null) return left;
    Node right = fns( root.right, t);
    if (right != null) retn
    ret null;
}
```

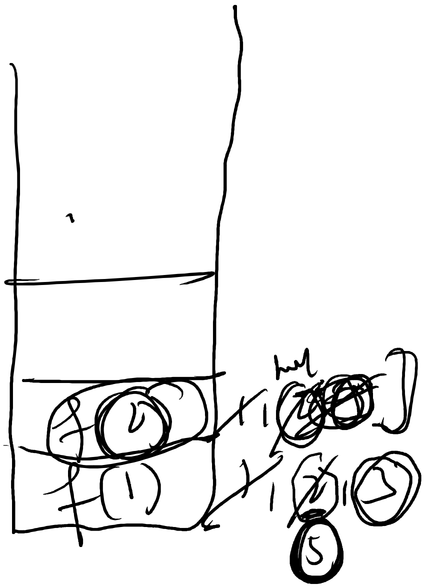


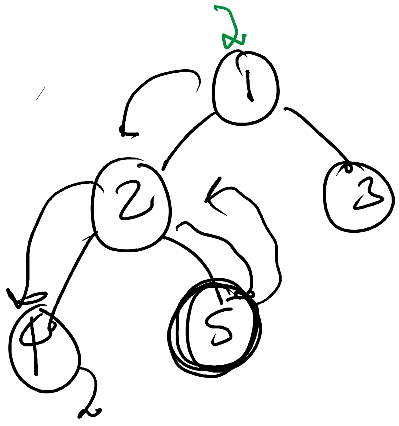
t= 11

t= 5



t=10





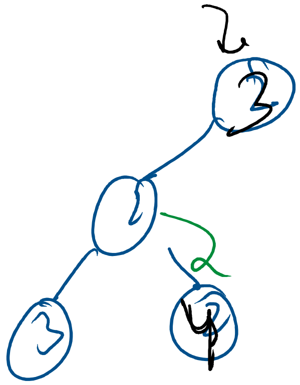
1 → record, null
2 → left
3 → right

[.

ren

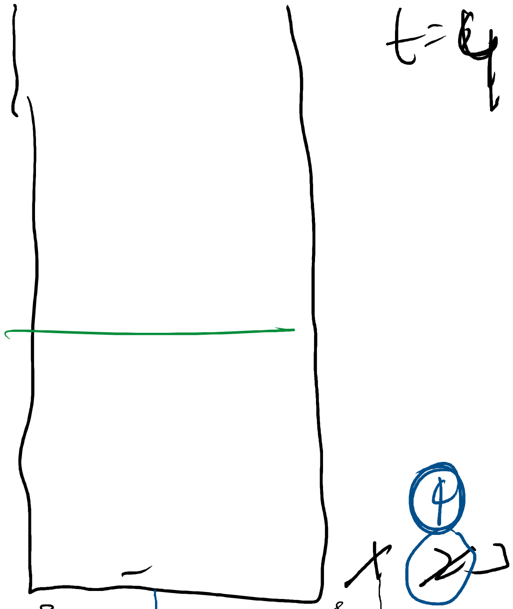
[5, 2, 1]

t=5



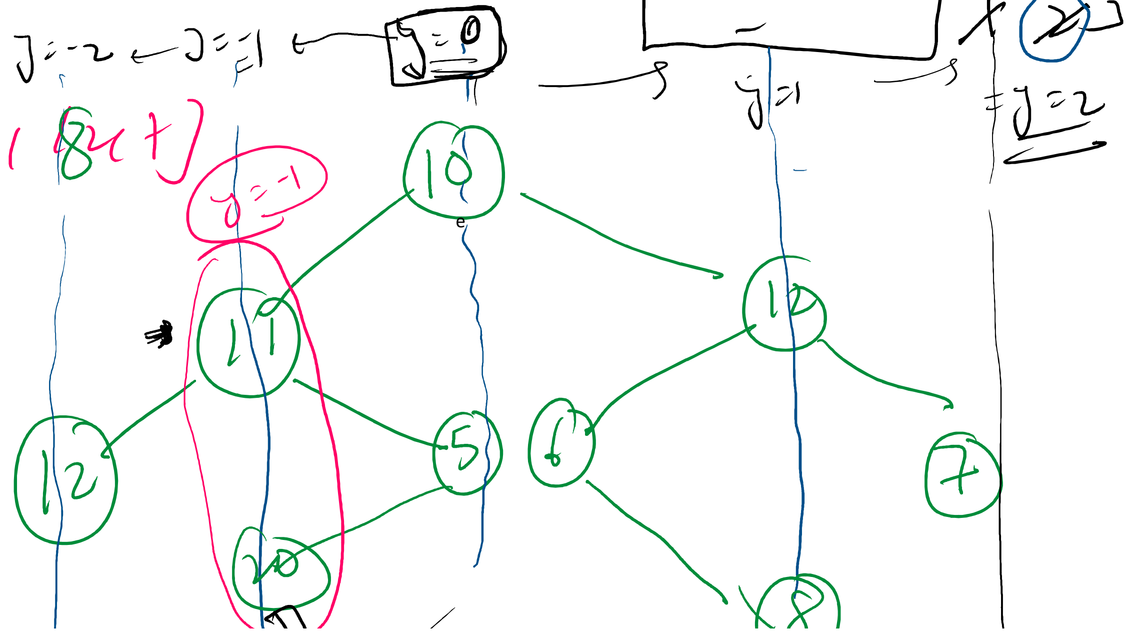
[4, 1, 3]

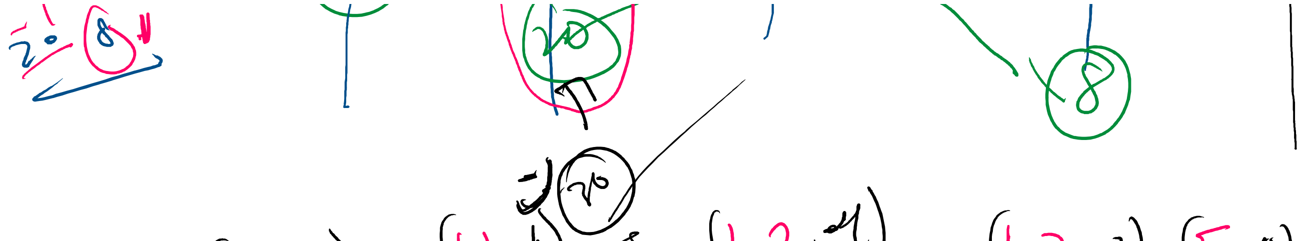
t=4



[1 2, 10 6, 8 2 7]

(10, 0)
(11, -1) 12, 0
12, -2 5, 0 6, 0 7, 2
2, 0 8



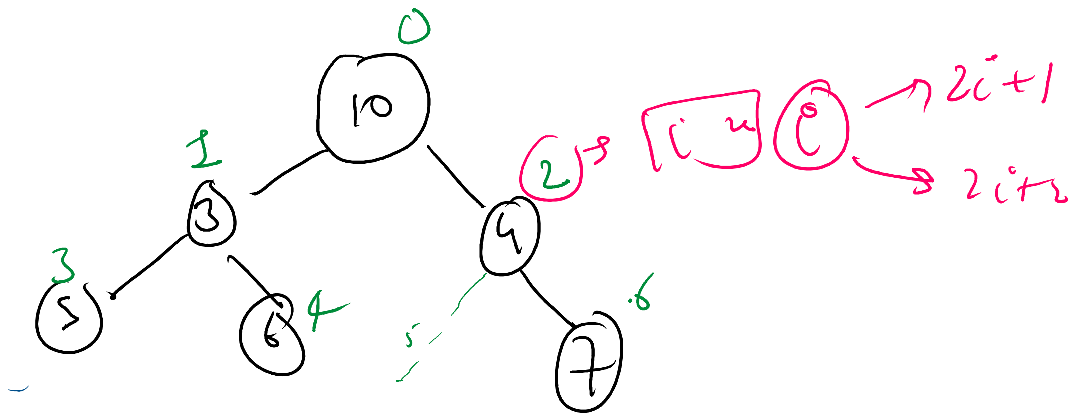


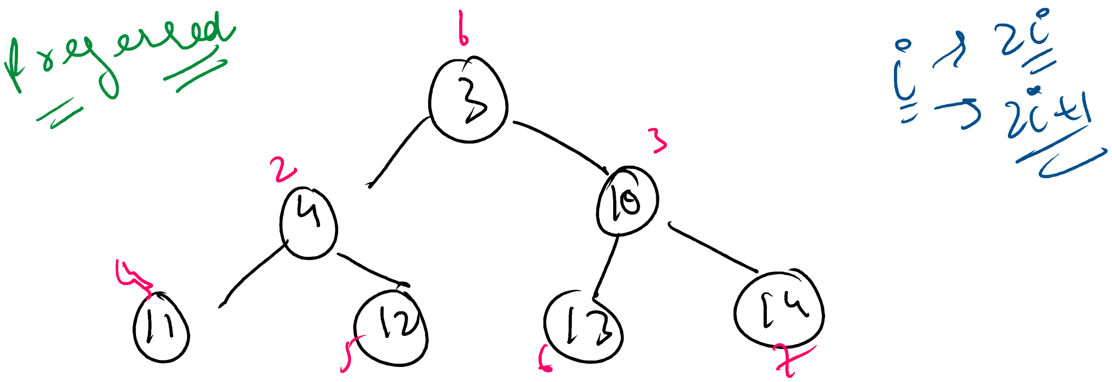
(10,0) (11,-1) (12,-1) (12,-2) (5,0) (6,0) (7,2)

(20,1) (8,1)

y	Node
0	10
-1	11
1	12
-2	12

#

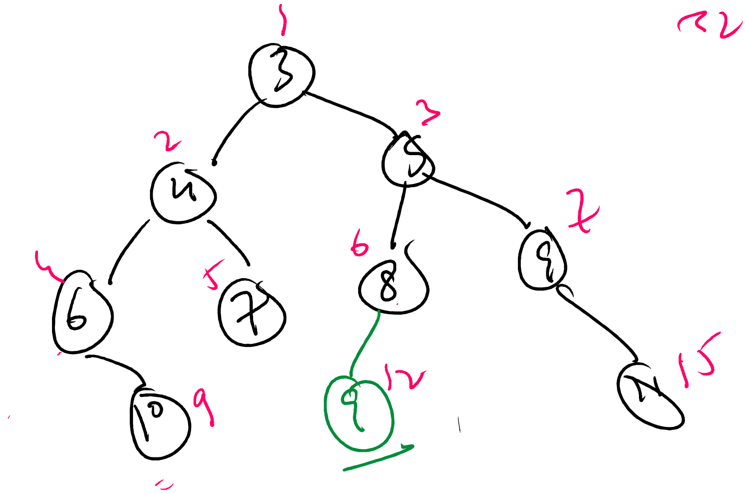


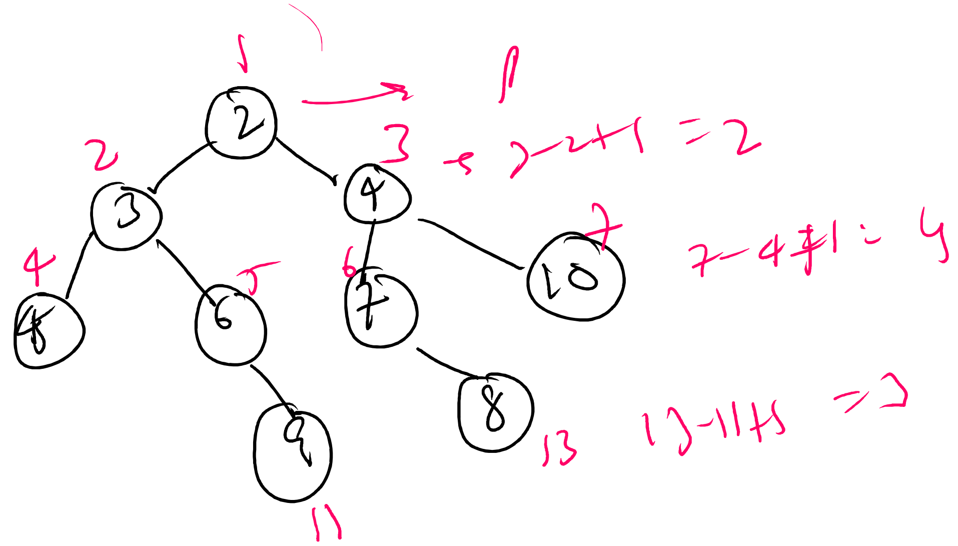


Width of a Tree

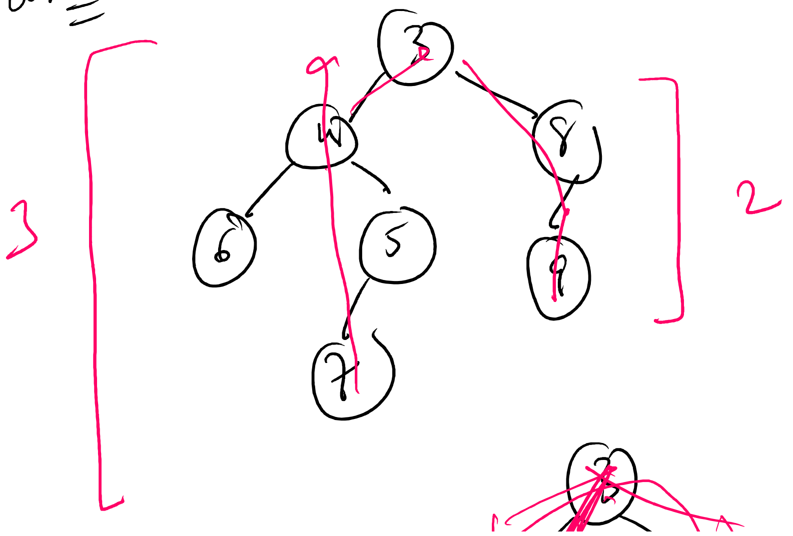
→ 2i

→ 2i+1





Diameter

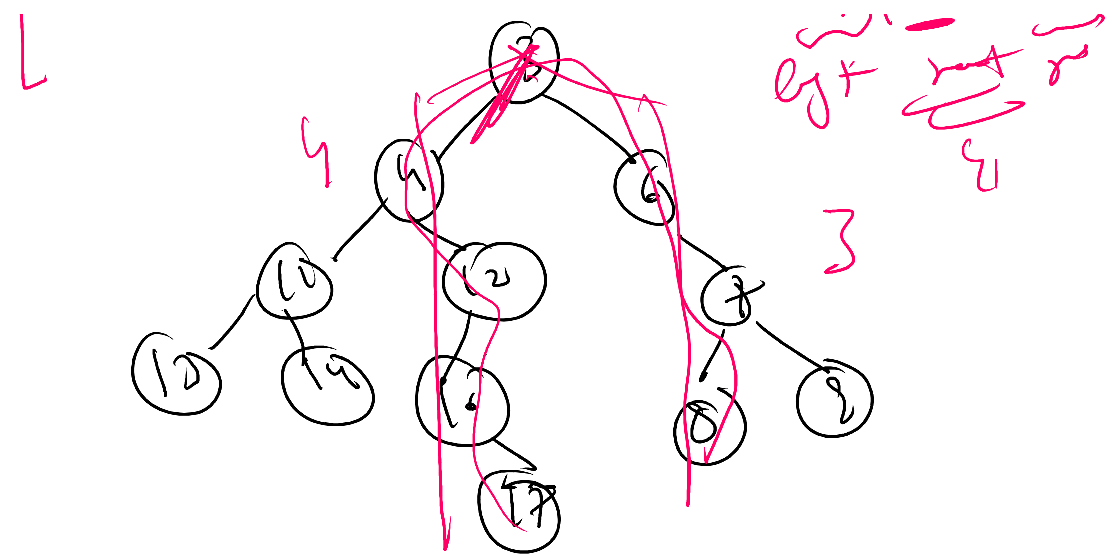


3+2+1

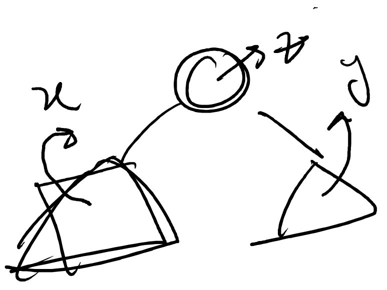
= 6

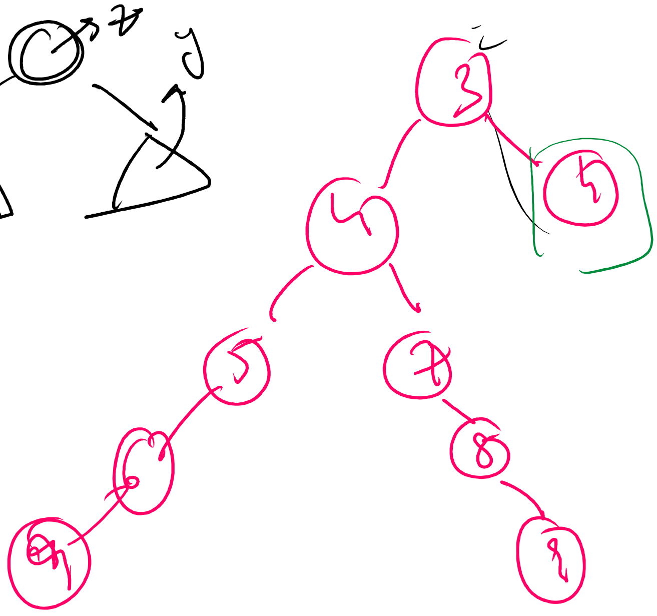
4+1+3

l + root + r



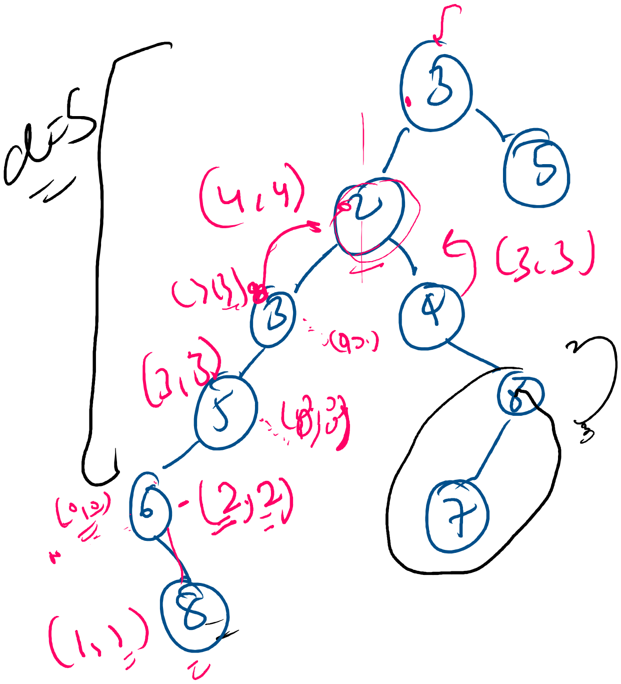
```
dia meter ( root) {
  If( root == null ) ret 0;
  int lh = height ( root.left);
  int rh = heigh ( root.ri)
    ret lh + 1 + rh;
)
```





```
dia( root) {
1 → if(root == null) ret 0;
2 → x = dia( root.left)
3 — y = dia( root.ri)
4 → z = h(root.l) + 1
          + h(root.r)
5 → } ret (x, y, z)max
}
```

```
diameter (root) {
  if(root==null) ret 0;
  int x == diamete(root.left);
  int y= diameter(root.right)
   int z = h(root.l)+1 + h(root.right)
      ret max(x,y,z);
}
```


dfs
3
5
(4,4)
2
(3,3)
(2,3)
3
4
(0,2)
(3,3)
5
(0,3)
8
6
(2,2)
(0,1)
7
(1,1)
8