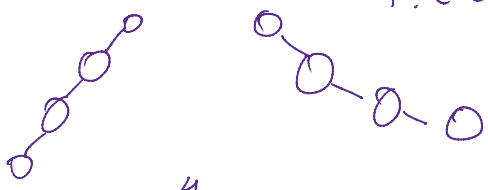
29 Jan

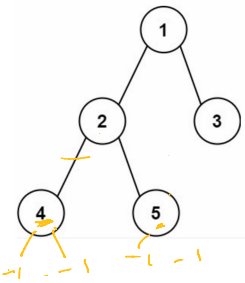
29 January 2022 20:09

T.C of calc. height = O(n)



4

Diameter of B.T :-

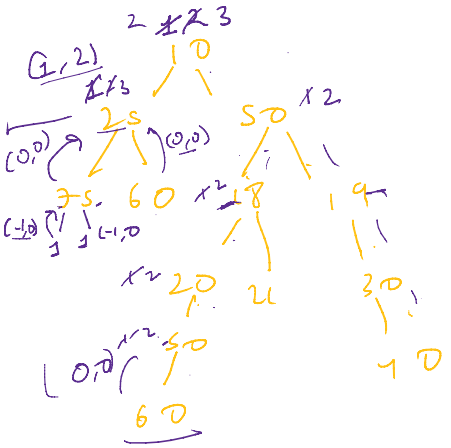


(n^2)

d = l+h+2

Pair of
int h;
int d;
}

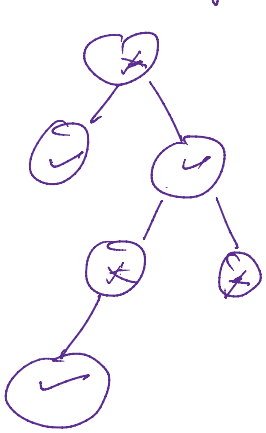
```
public static Pair diameterPair(Node node) {
1  if (node == null) return new Pair(h: -1, d: 0);
2  Pair l = diameterPair(node.left);
3  Pair r = diameterPair(node.right);
4  int currDia = l.h + r.h + 2;
   int h = Math.max(l.h, r.h) + 1;
5  return new Pair(h, Math.max(currDia, Math.max(l.d, r.d)));
}
```

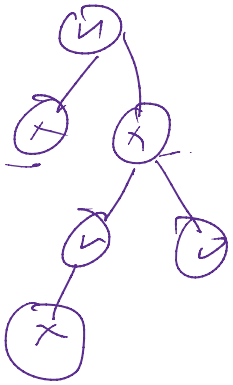


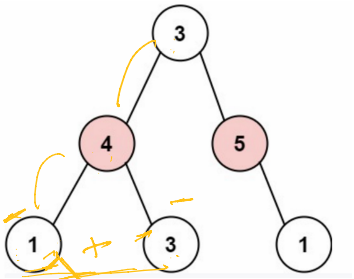
3 + 2 + 2 = 7

(4, 3) + 1 = 5

House Robbery in BT :-





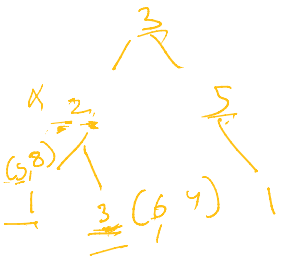


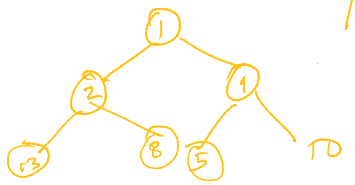
Pair of
rob :
noRob;
}

Pair sol (root)
if(root==null) return Pair(0, 0);

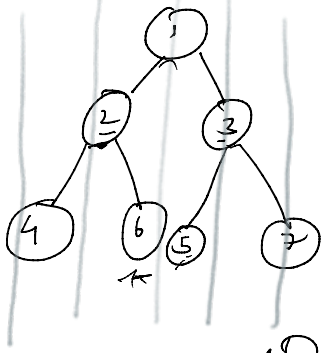
l = sol(root.left)
r = sol(root.right);
currRob = root.val + l.noRob + r.noRob;
currNoRob = Math.max(l.rob, l.noRob) + Math.max(r.rob, r.noRob);

```
public static RobberyPair maxRobbery(Node root) {
    if (root == null) return new RobberyPair( rob: 0,  noRob: 0);
    RobberyPair l = maxRobbery(root.left);
    RobberyPair r = maxRobbery(root.right);
    int currRob = root.data + l.noRob + r.noRob;
    int currNoRob = Math.max(l.noRob, l.rob) + Math.max(r.rob, r.noRob);
    return new RobberyPair(currRob, currNoRob);
}
```

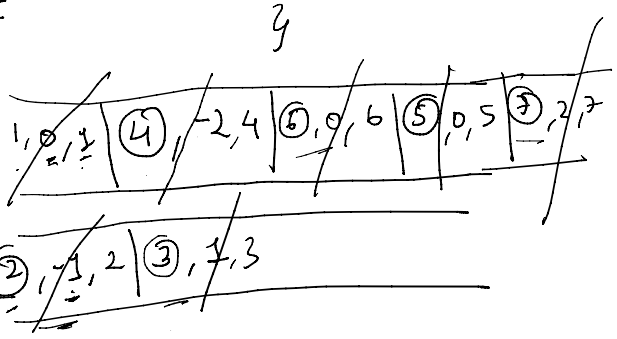




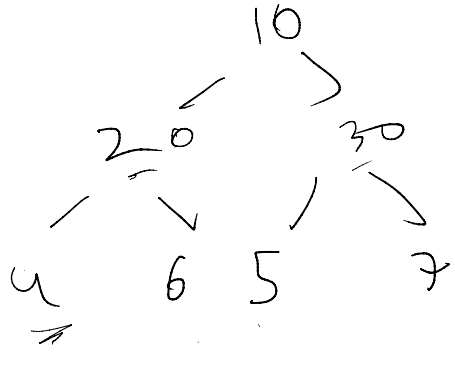
Vertical Orders Traversal with Priority :-

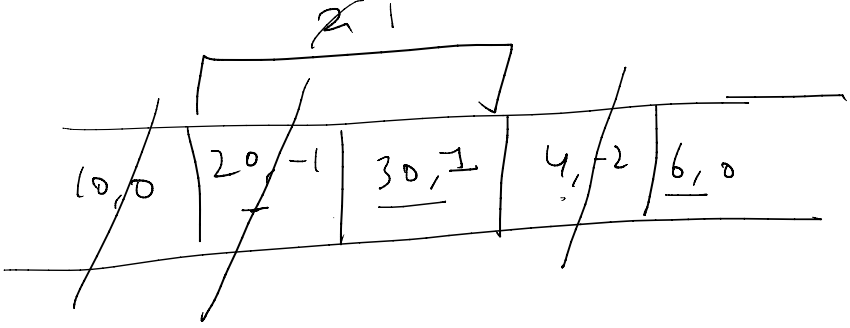


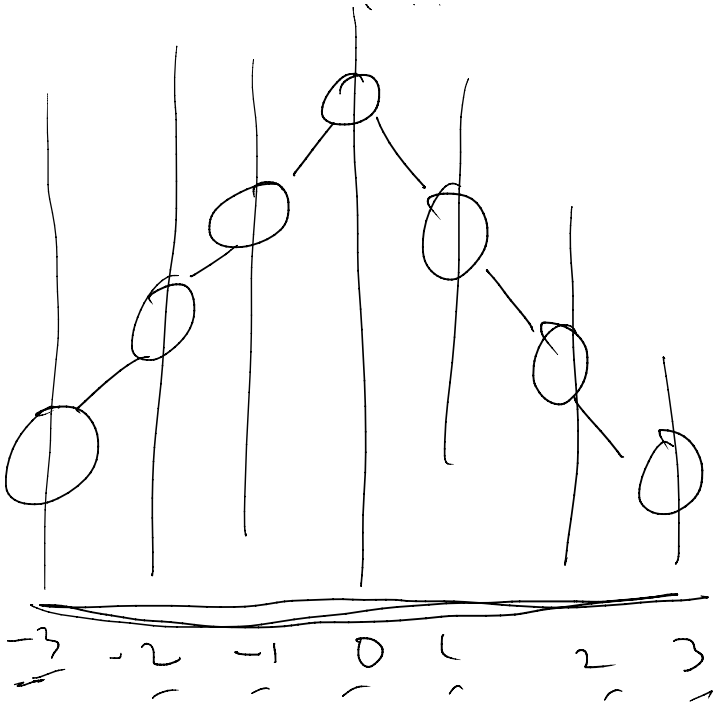
Pair {
Node n
int level
int data;
}



Map
Int — list
0 — [1, 5, 6]
-1 — [2]
1 — [3]
-2 — [4]
2 — [7]







non-min+1 =

$l = 2 \times i + 1$

$r = 2 \times i + 1$