

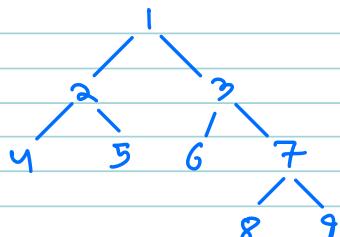
Trees 5 : Problems On Trees

TODAY's CONTENT

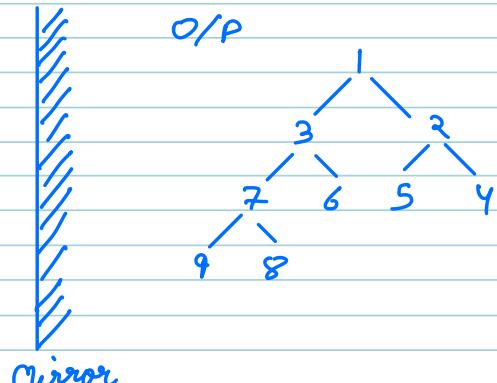
- Invert Binary Tree
- Equal Tree Partition
- Next Counter Binary Tree
- Root to leaf Path sum = K
- Diameter of Tree

Question :- Given the root node of a Binary Tree,
Write a function to invert the tree

I/B

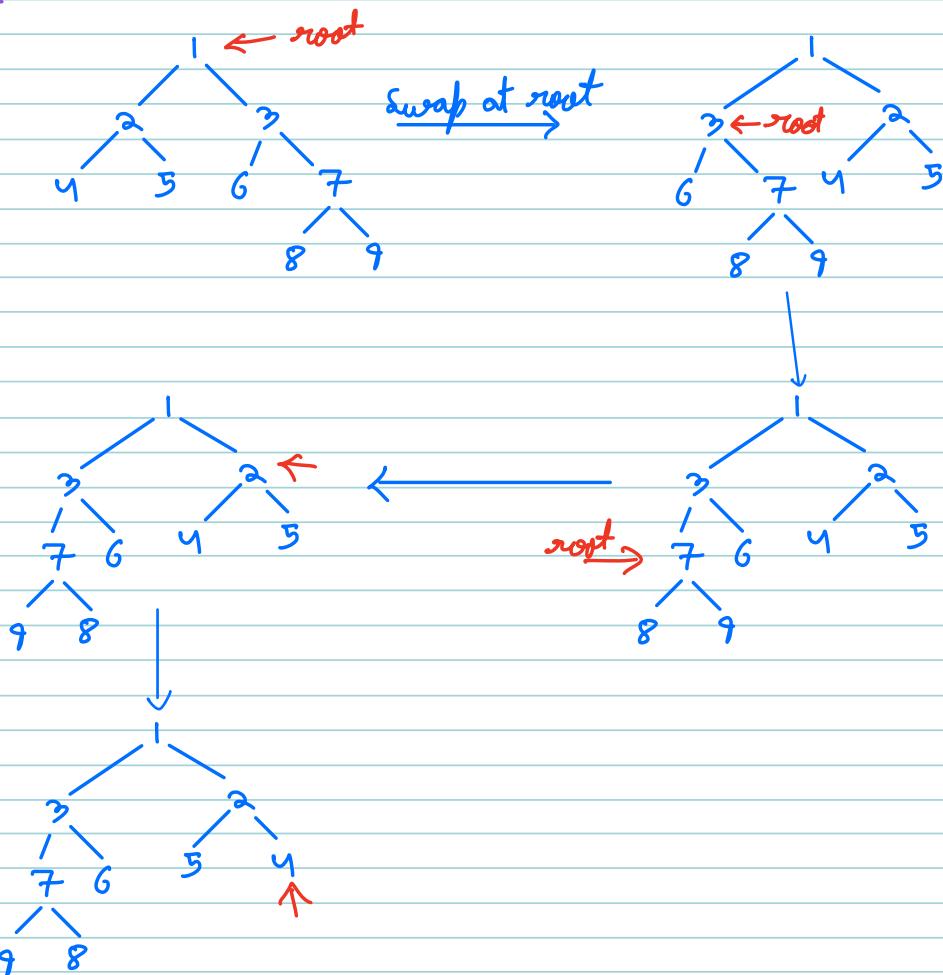


O/P



Mirror

IDEA



Conclusion :- + all the Nodes Swap Left & Right

PSEUDO CODE

```
void invert ( Node root ) {  
    if ( root == Null ) { return; }  
    temp = root . Left ;  
    root . Left = root . Right ;  
    root . Right = temp ;  
    invert ( root . Left );
```

|
}|
invert (root · right);

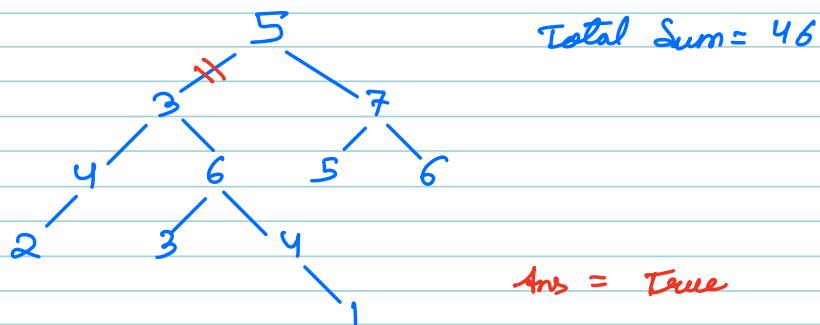
$$\boxed{T \subset \rightarrow O(N)} \\ S \subset \rightarrow O(H)}$$

Question :- Equal Tree partition

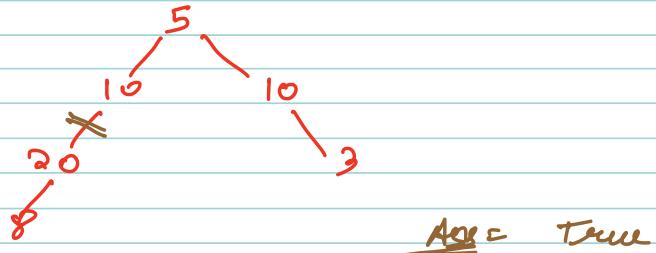
data >= 1

check if it is possible to remove an edge from Binary Tree such that the sum of resultant two tree is equal.

Ex



Ques 1 :- check whether the given tree can be split into two non empty subtrees with equal sums or not.



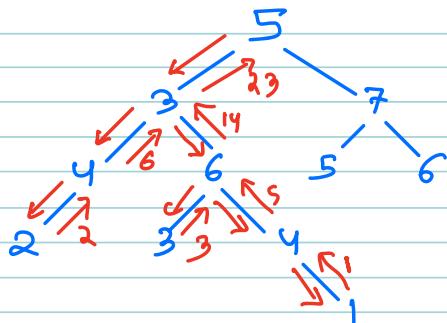
OBSERVATION

① If total sum of the Tree is S , then the subtree would have sum of $S/2$

② If total sum is odd, then return False;

APPROACH:- Check if there exists a subtree with
Sum = $\frac{\text{Total Sum}}{2}$

DRY RUN



PSEUDO CODE

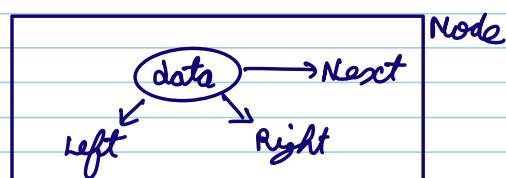
```
S = sum (root); // TODO
if (S % 2 == 1) {
    return False;
}
boolean ans = False;
int check (Node root) {
    if (root == null) {return 0;}
    L = check (root.left);
    R = check (root.right);
    if (L == S/2 || R == S/2) {
        ans = True;
    }
    return L + R + root.data;
}
```

TC = $O(N)$

SC = $O(1)$

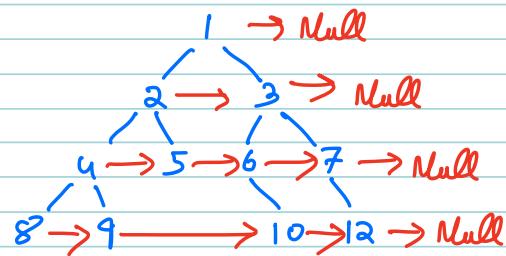
Question:- Next Pointer in Binary Tree

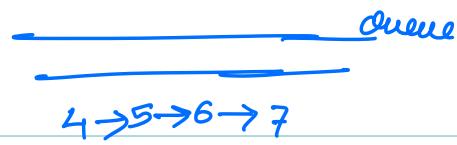
Initially Each Node's next pointer points to null.



Update Each Node's next pointer to point to the next node in same level.

Ex :-





PSEUDO CODE

void levelOrder Traversal (Node root) {

 queue < Node > q;

 q. add (root);

 while (q. size () > 0) {

 int N = q. size ();

 for (i=1 ; i<=N ; i++) {

 Node front = q. peek ();

 q. remove ();

 if (i != N) { front. next = q. peek (); }

 if (front. left != null) {

 q. add (front. left);

 if (front. right != null) {

 q. add (front. right);

}

}

TC = O(N)

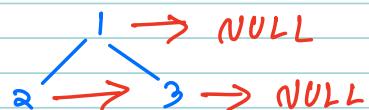
SC = O(N)

Follow up :- Fill Next in perfect Binary Tree

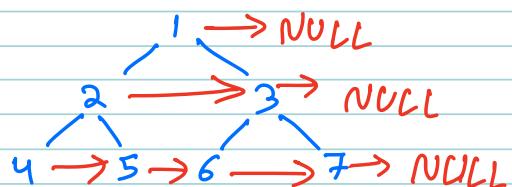
[Expected SC $\rightarrow O(1)$]

```
class Node {  
    int data;  
    Node left, right, next;  
    Node(x) {  
        data = x;  
        Left = null;  
        Right = null;  
        next = null;  
    }  
}
```

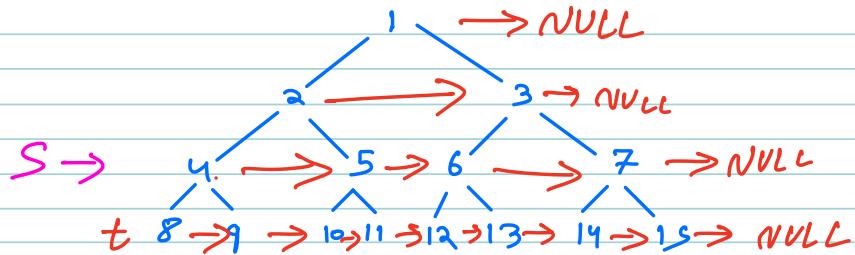
Ex 1 :-



Ex 2 :-



IDEA



$t.\text{Left}.\text{next} = t.\text{right}$

if ($t.\text{next} \neq \text{null}$) {

$t.\text{right}.\text{next} = t.\text{next}.\text{left}$

}

$t = t.\text{next}.$

Q How will we move to next level?

$\text{while } (t.\text{left} := \text{null}) \{$

$\text{Node } S = t;$

$\text{while } (+ := \text{null}) \{$

$t.\text{Left}.\text{next} = t.\text{right}$

if ($t.\text{next} \neq \text{null}$) {

$t.\text{right}.\text{next} = t.\text{next}.\text{left}$

}

$t = t.\text{next}.$

}

$t = S.\text{Left};$

}

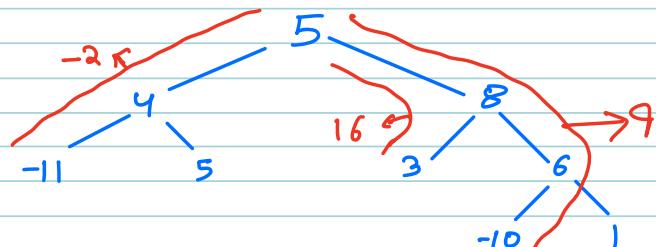
$TC \rightarrow OCN$
 $SC \rightarrow OCD$

Question :- Given a binary tree & an integer K , determine if there exists a root to leaf Path in tree such that adding up all the node values along the path equals K .

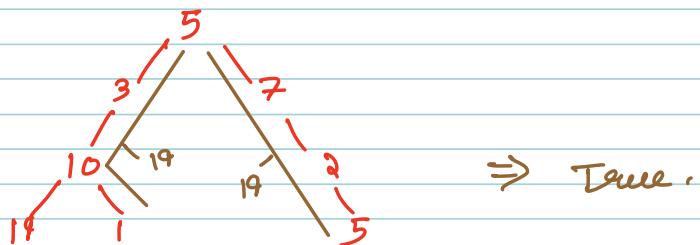
$K = 16 \rightarrow \text{True}$

$K = -2 \rightarrow \text{True}$

$K = 9 \rightarrow \text{True}$

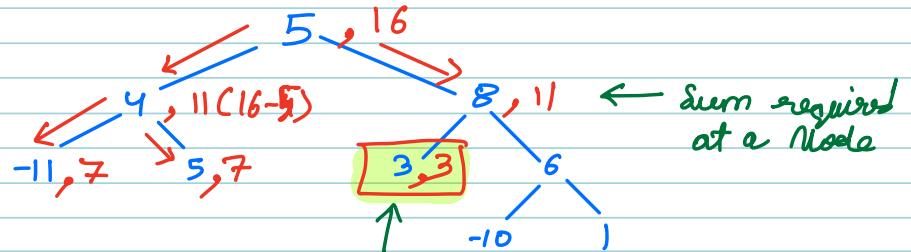


Ques 2 :- Tell if there exist a root to leaf path with sum value $K = 19$



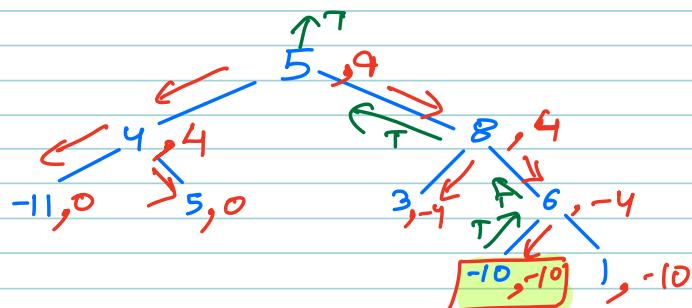
IDEA:-

$K = 16$



It Should be Leaf &
Node.value == required sum

$\underline{\underline{K = 9}}$



PSEUDO CODE

```
boolean check( Node root, int k){
```

```
    if (root == NULL) { return False; }
```

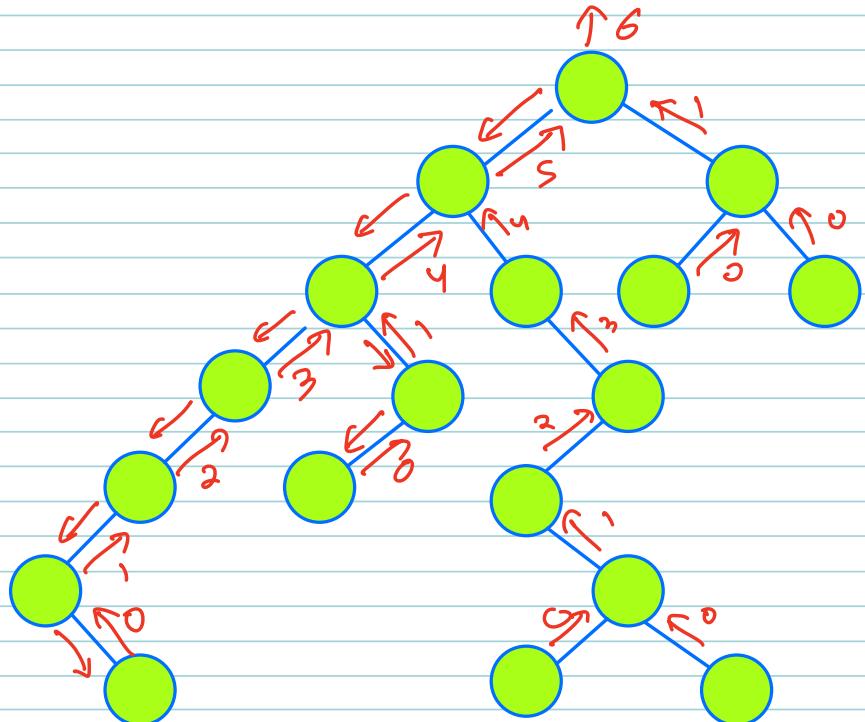
```
    if (root.left == Null && root.right == null)
```

```
        return (root.data == k);
```

TC - O(N)
SC - O(H)

```
    return check(root.left, K-root.data)  
    || check(root.right, K-root.data);
```

question :- Height of a BT In terms of edges



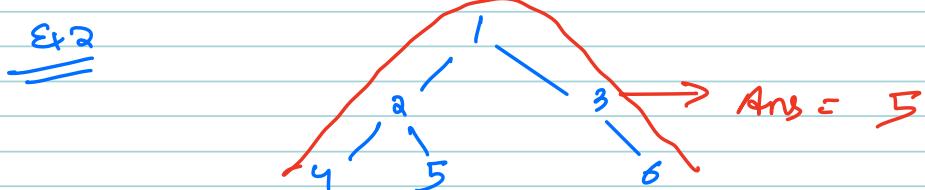
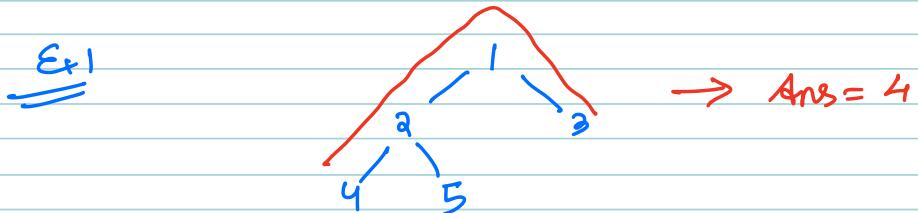
PSEUDO CODE

```
int height ( Node root )
{
    if ( root == null ) { return -1; }
    LH = height ( root.left );
    RH = height ( root.right );
    return max ( LH, RH ) + 1;
}
```

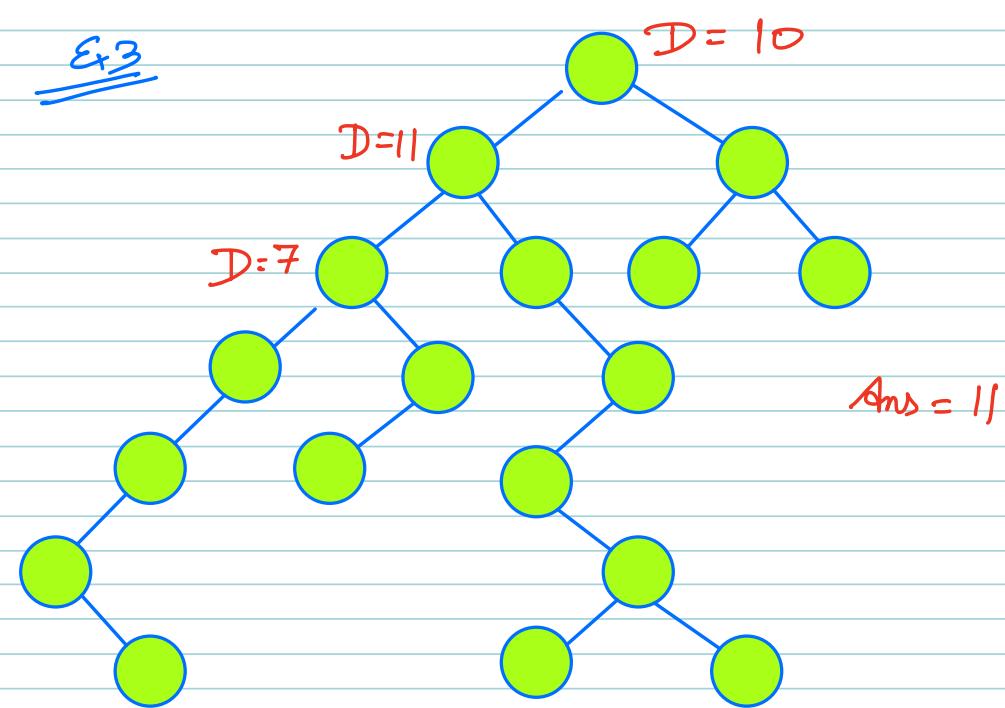
TC - OCN
SC - O(H)

Question :- Diameter of tree

Diameter :- It is the no. of Nodes on longest Path b/w two nodes in a binary Tree.



Ex 3



Learning :- It is not important to include root.

Ques 3 :- How would you find the diameter of a binary tree?

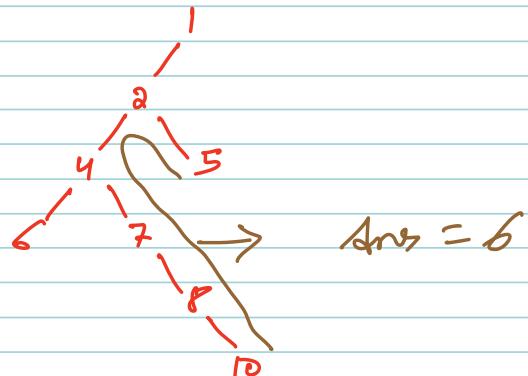
↳ Max (Diameter of LST, Diameter of RST,

Sum of Left Height + right height plus)
one

OBSERVATION :- Diameter through $\Rightarrow LH + RH + 1$

In terms of nodes

Ques 4 :- Diameter of the given tree



PSEUDO CODE

```
int diameter = 0;  
int findDiameter (Node root){  
    if (root == null) { return 0; }  
    LH = height (root.left);  
    RH = height (root.right);  
  
    diameter = Max(diameter, LH+RH+1);  
    return Max (LH, RH) + 1;  
}
```

TC - OCN
SC - OCH