26-Jan

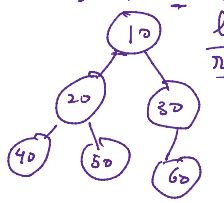
26 January 2022 19:11

Node {
int data
Node left;
Node right;
}

int[] values;

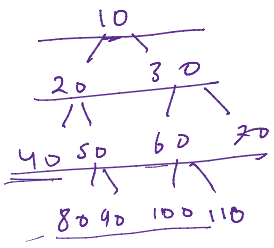
{10, 20, 30, 40, 50, 60} i = i+2

0 1 2 3 4 5

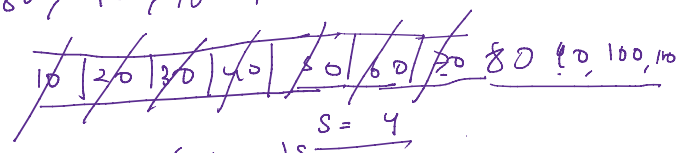


l = 2xi+1
r = 2xi+2

BFS :- Breadth First Search



10, 20, 30, 40, 50, 60, 70
80, 90, 100, 110

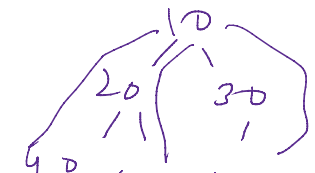


S = 4

10, 20, 30, 40, 50, 60, 70,

```
while(s!=0){
Node n = q.remove();
sout(n);
if(n.left!=null) q.add(n.left);
if(n.right!=null) q.add(n.right);
s--;
}
```

All paths :- https://leetcode.com/problems/binary-tree-paths/submissions/



"10"

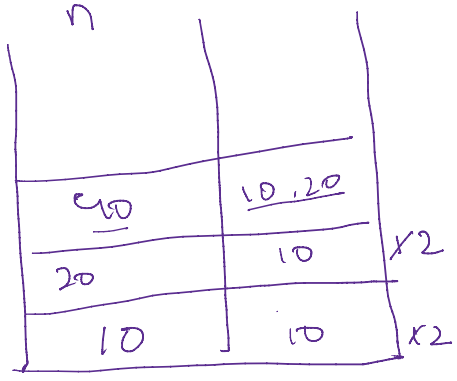
getAllPath(n, paths, "10")

① if(n.left == n.right == null) {
list.add(path+n) return;

② if(n.left != null) {
getAllPath(n.left, path+n);
}

③ if(n.right != null) {
getAllPath(n.right, path+n);
}
}

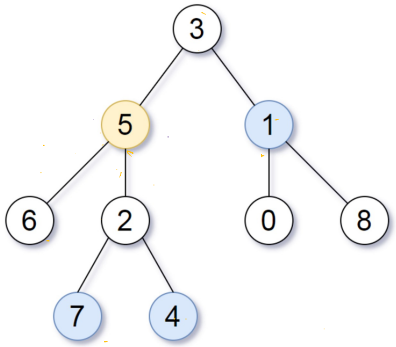
path



All nodes distance k in BT :- https://leetcode.com/problems/all-nodes-distance-k-in-binary-tree/

target = 5

k = 2



Node one

k = 2

allnodesk (n, k, tar, ans) {

if (n == 0) return 0

if (n.val == tar) {

allnodesk down (n, k, ans);

return 1;

}

int res = allNodesk(n.lft, k, tar, ans)

res = res > 0 ? allNodesk(n.right, k - res - 1, tar, ans) :

allNodesk (n.right, k, tar, ans);

return res > 0 ? res + 1 : res;

}