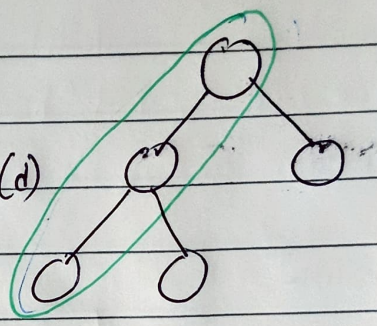
222: Count nodes in Complete Binary Tree

Do it in less the $O(n)$, no obvious solution.

The number of nodes at the last depth can be from the range $1 \longleftrightarrow 2^{d-1}$ (From the left side)

(1) My solution



- Get the height from the left side(d)
- Count the leaves from the left,
  if you find a leave that is not at the depth d calculated above, stop.
  then return 2^{d-1} + leaves − 1

Worst case $O(n)$ but still better.

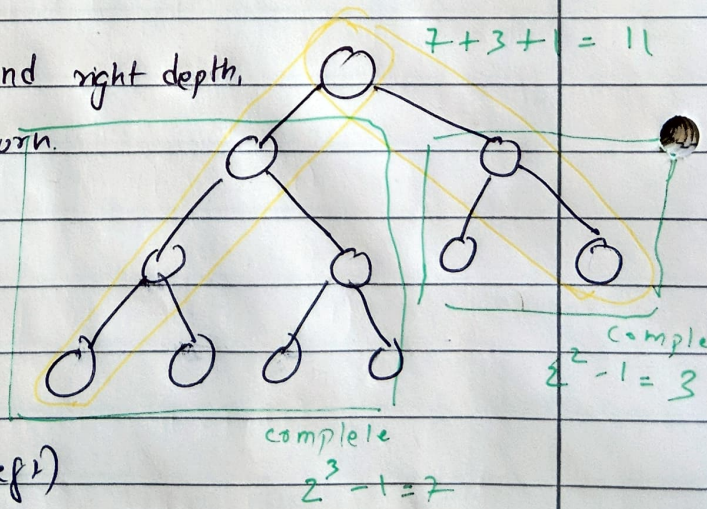
(2) Best solution

→ Find the left depth and right depth,
if they are equal return.

(complete tree) $2^d - 1$

else:

(if not complete)




```
leftcount = func(root → left)
rightcount = func(root → right)
return 1 + leftcount + rightcount
```