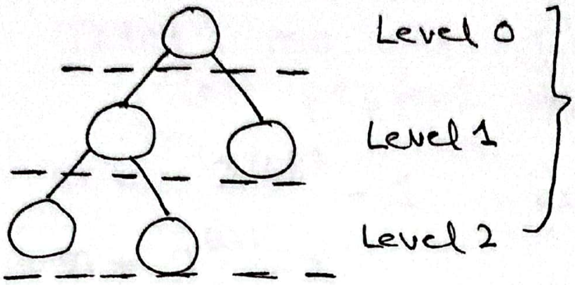
In order → Root in b/w left & right
pre order → Root before " "
post order → Root after " "

① Strict /Proper Binary Tree:

If binary tree's each node can have either 2 or 0 children nodes.
like:

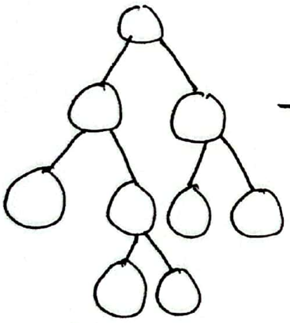


Depth of Tree = 3
↑
Max level of node

② Complete Binary Tree:

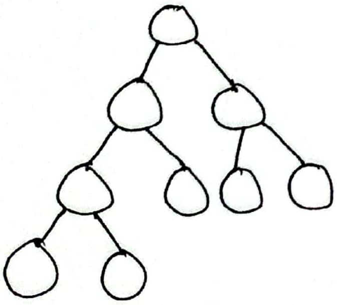
All levels except possibly the last one are completely filled & all nodes are as left as possible (In last level if not compeletly filled).

e.g.:



→ Not a complete Binary Tree as left most place is empty in last level.

⇓

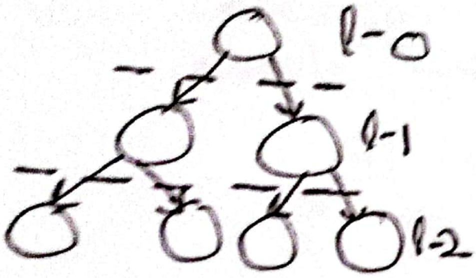


→ complete Binary Tree:

↳ If last level is also filled It is known as Perfect binary Tree.

so in case of Perfect Binary Tree:

Every level will have → 2^l nodes
where l → level.



→ 2^0 → 1 node

→ 2^1 → 2 nodes

→ 2^2 → 4 nodes

Total no of nodes
in perfect binary tree:

$n = 2^{h+1} - 1$ where h = height ↳ Max leve

$n = 2^{2+1} - 1$

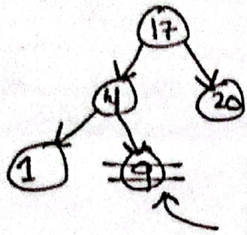
$n = 2^3 - 1$

$\boxed{n = 7}$

(ii) Binary Tree | BST (Part 2)

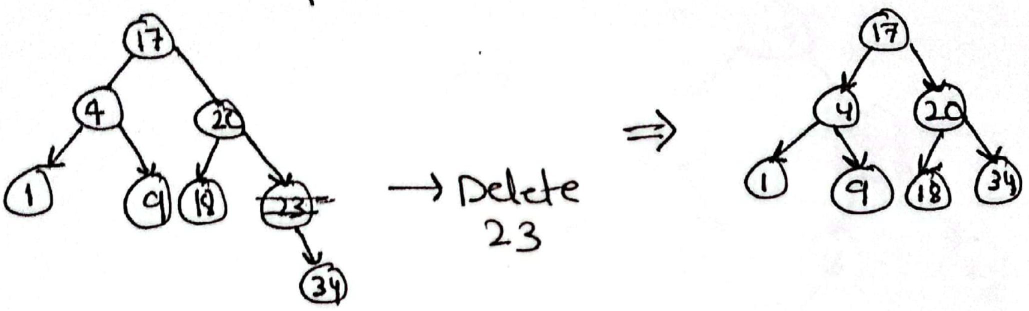
Deleting a Node from Binary Tree:

(i) Deleting Node with no child is simple → Just delete the node



9 deleted

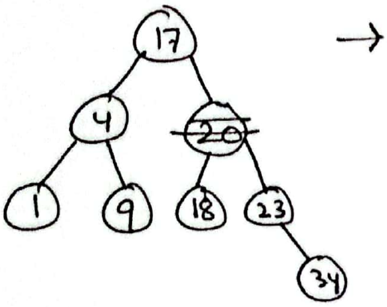
(ii) Deleting Node with one child is also simple.



→ Delete 23

← Move the other element in that place.

(iii) Deleting Node with Two childs:

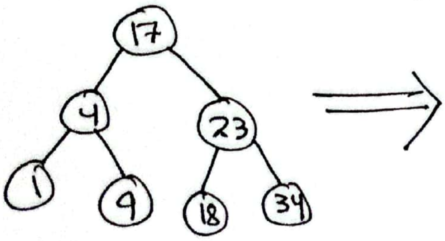


→ If you want to delete 20 you have to take care of property of BST.

less values → left side

greater " → right side.

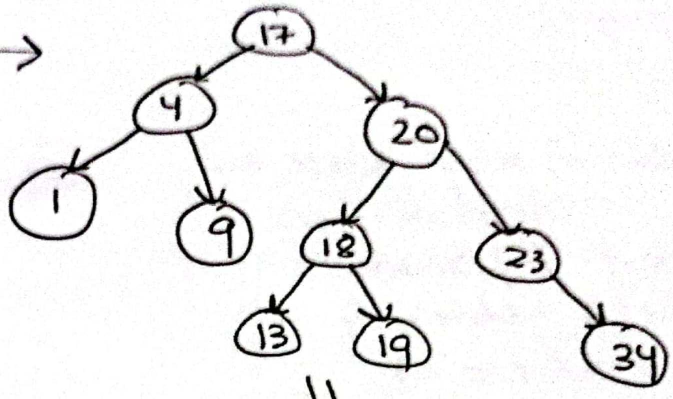
(i) So first technique is find minimum values on right and move that to place of node you removed.



Scanned with CamScanner

→ Other Technique is To find maximum node from right subtree, copy it to place where you removed node & then delete that node from previous location (cut & paste)

→

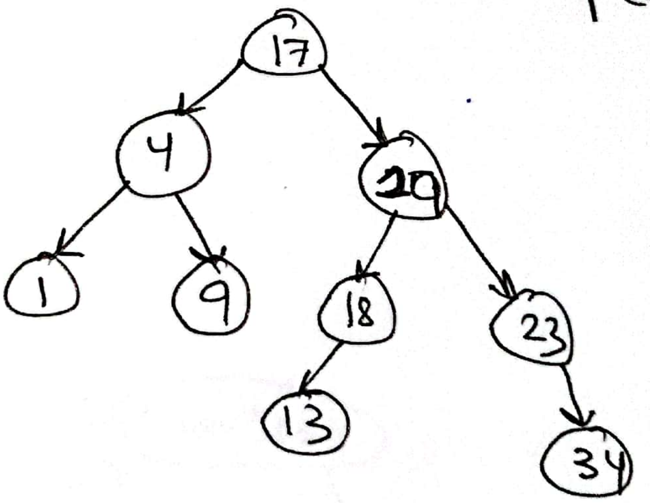


⇓

if you want to delete 20 :

→ Max in right subtree is 19 move it in its place.

⇒



Scanned with CamScanner