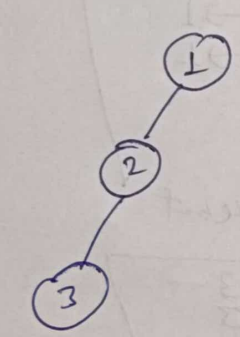
Que. Can you construct a unique Binary Tree with following?

(a) Preorder & postorder

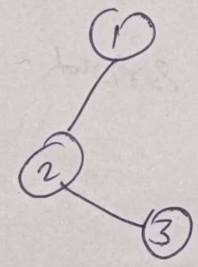
ex

pre-order → 1 2 3

post-order → 3 2 1



pre-order – 1 2 3
post-order – 3 2 1



pre-order – 1 2 3
post-order – 3 2 1

from preorder & postorder you can't construct a unique BT.

(b) Inorder & Preorder

Yes

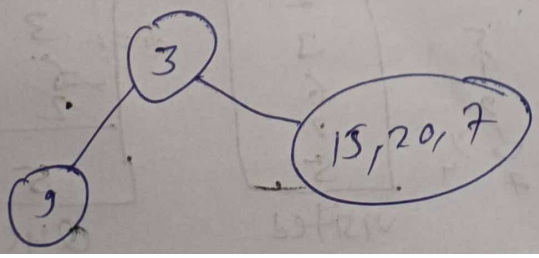
left root right

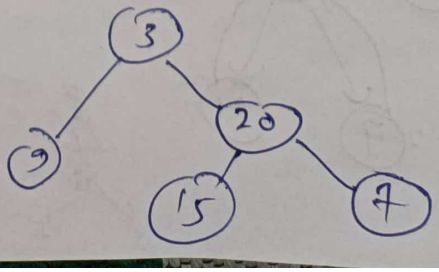
Inorder = [9, 3, 15, 20, 7]

root

preorder = [3, 9, 20, 15, 7]

root left right





(c) Inorder & Postorder → Yes

Construct a Binary Tree from in & pre order Traversal

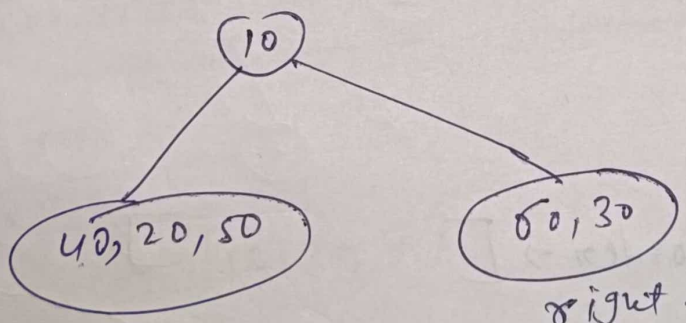
(Left root right)

Inorder → [40 20 50 10 60 30]

preorder → [10 20 40 50 30 60]

(root left right)

root of the tree



left sub tree

inorder → [40, 20, 50]

preorder → [20, 40, 50]

Now we have same problem like we are given in & pre order & we have to generate a BT.

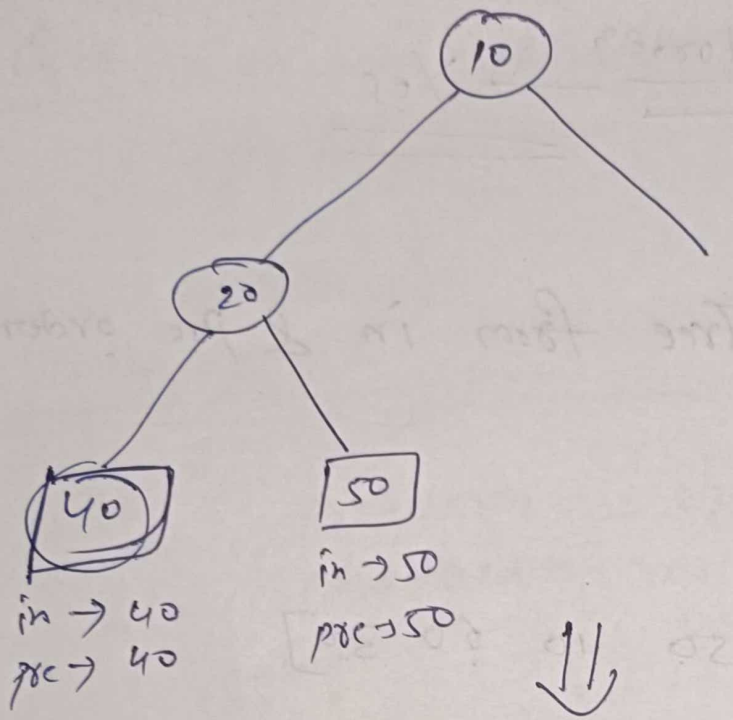
right subtree

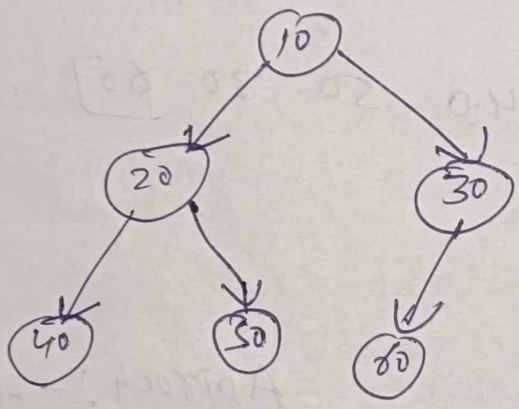
in → [60, 30]

pre → [30, 60]

Approach :- for every recursive call we get the root from pre order & left from inorder right from inorder

Now when we call for left subtree then we pass the in order & preorder for left subtree also similiarly for the right subtree





How to code this

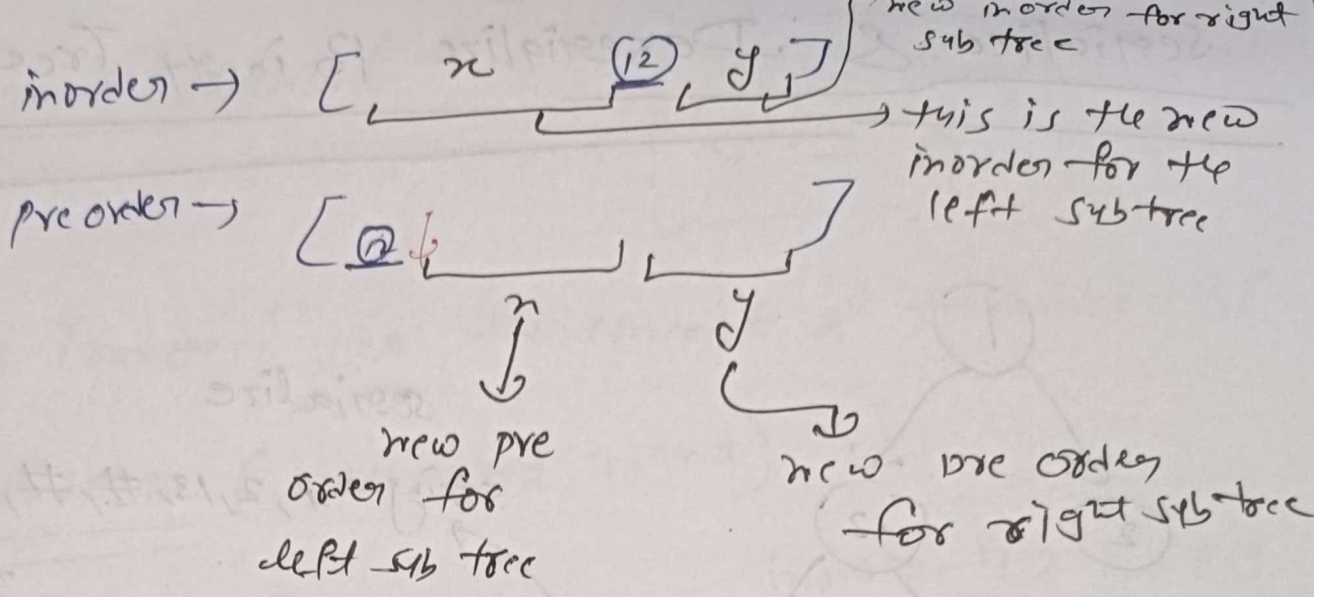
Step: 1

inorder → [ ]

preorder → [ ]

Hash this (val, idx)

every first element of preorder will be root of that tree



T.C. = $O(N)$

S.C. = $O(N) + O(N)$

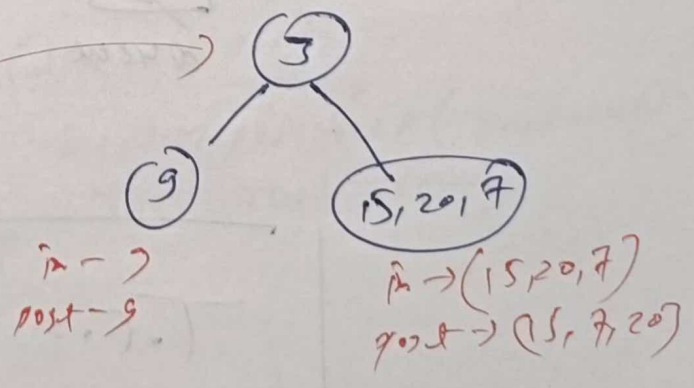
↑ map ↑ recursive stack

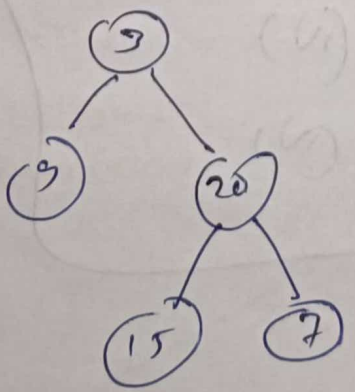
Construct BT from inorder & postorder

Same as previous

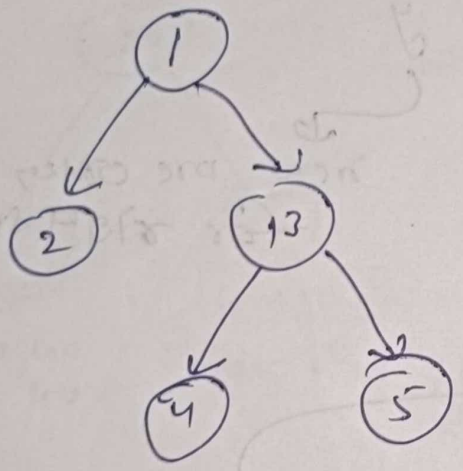
in → [9, 3, 15, 20, 7]

post → [9, 15, 7, 20, 3]





Serialize & Deserialize Binary Tree

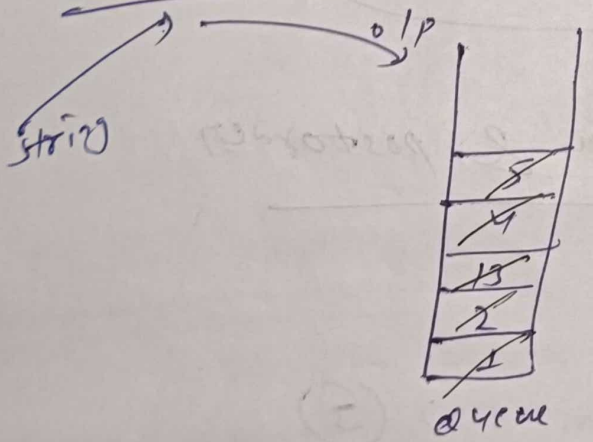


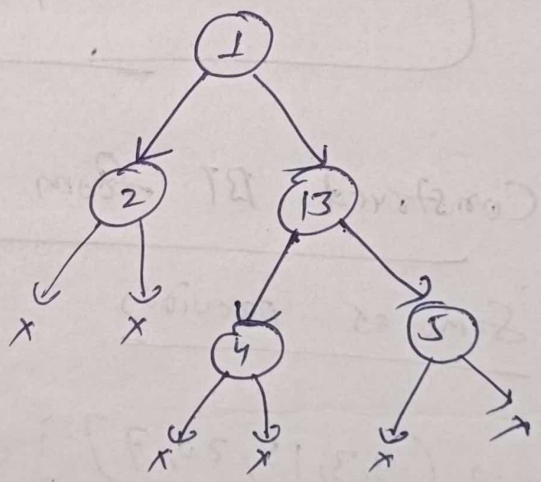
serialize

string → 1, 2, 13, #, #, 4, 5, #, #, #, #,

level order traversal.

Deserialize





T.C. = O(N)

S.C. = O(N)