Homework 2

10/14/2024

2.) Given: I (inorder); P (Preorder); n (nodes); T (binary Tree); A (Arbitrary node)

base case: $n=1$ let: $I=[A]$; $P=[A]$; $T=A$.

root inorder Pre order

$R=P[0]$ $I=[L\ R\ R']$ $P=[R\ P_L\ P_{R'}]$

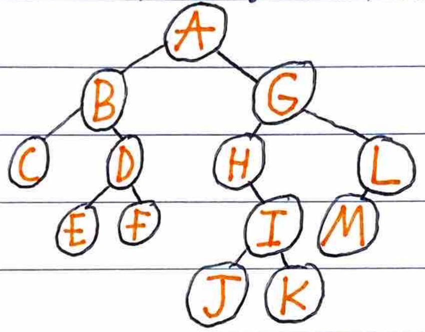
where $|P_L|=|L|$ & $|P_{R'}|=|R'|$

Left subtree: (L, P_L) righ subtree: $(R', P_{R'})$

By induction, for all $n\geq 1$, there exist T from sequences I&P.

3.) Draw binary tree, Preorder: ABCDEFGHIJKLM

inorder: CEDFBAHJIKGML



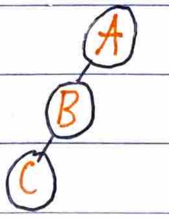
4.) Show a Preorder and Post order that isn't Possible for the same tree.

Preorder: A, B, C

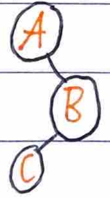
Postorder: C, B, A

Can't be the Same tree.

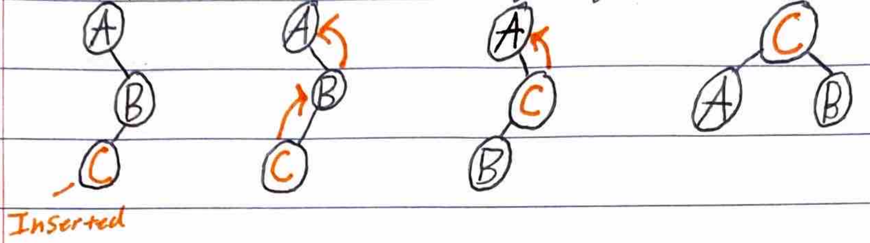
Pre order



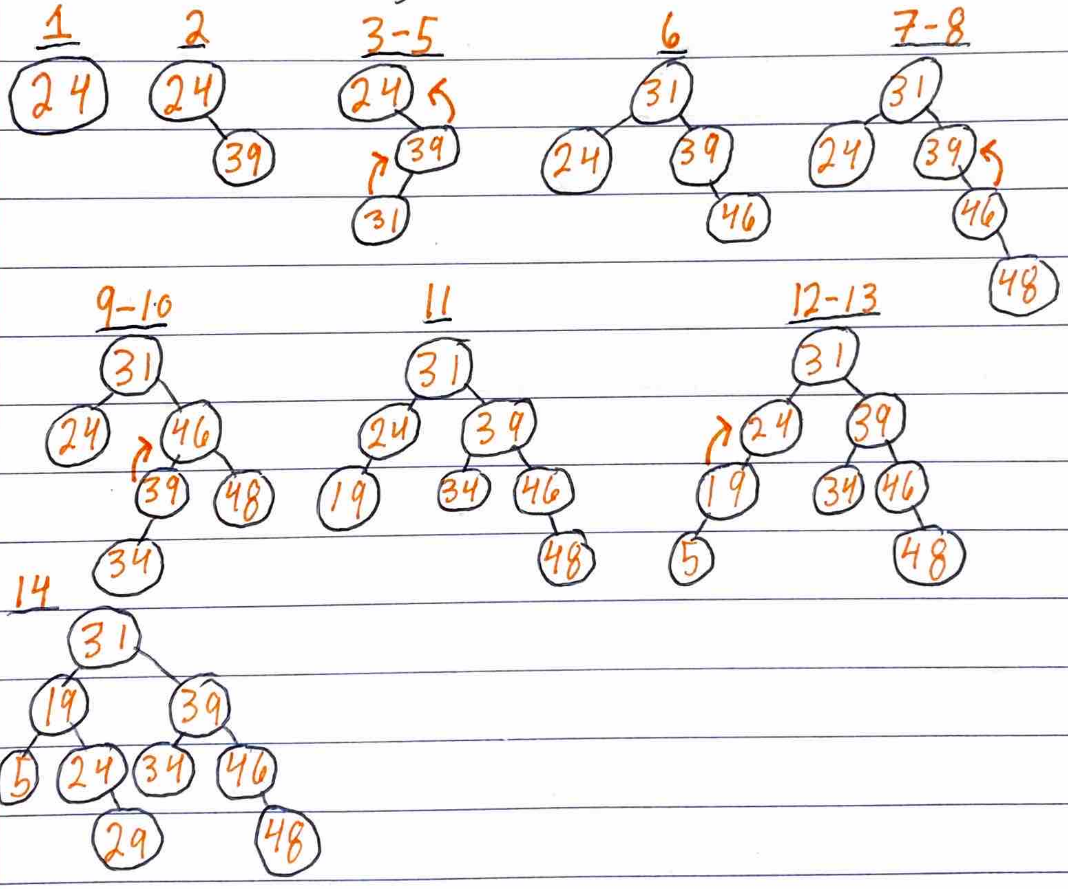
Post order



5.) AVL Insertion, needing right-left double rotation



6.) Put the numbers given into an empty AVL tree
given: 24, 39, 31, 46, 48, 34, 19, 5, 29



7.) What would be the complexity of constructing an n-node binary search trees?

If the array is properly sorted for the binary search tree big $O(\log n)$.

If the array isn't properly sorted for the binary search tree big $O(n^2)$.

It would only have to go through half of the elements since it's sorted. We won't have to iterate through every case for inserting, deleting or anything else.

But if it isn't sorted you have to go through every element.