

Topic : AVL Trees : Rotations

Recap:

AVL Trees : Height - Balanced Binary Search trees

Height of an n -node height - balanced tree is $\Theta(\log n)$.

Accessor operations on AVL trees.

List in sorted order $O(n)$

Search

Min, Max,

Predecessor, Successor

$$\left\{ \begin{array}{l} O(h) = O(\log n) \end{array} \right.$$

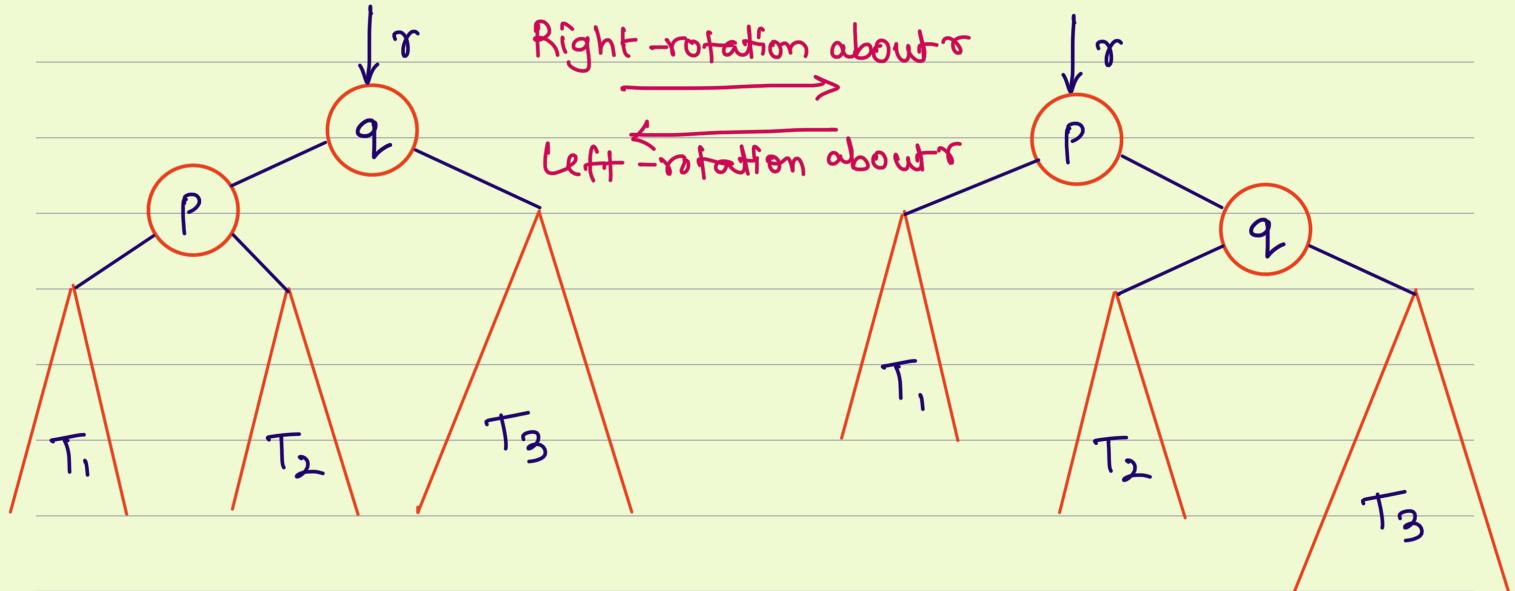
Exactly like on
Binary
Search
Trees!

Modifier Operations on AVL trees

- Add element (Insertion)
 - Remove element (Deletion)
- } 1. Insert / Delete like in a Binary Search tree
2. Fix imbalances
 ↳ How?

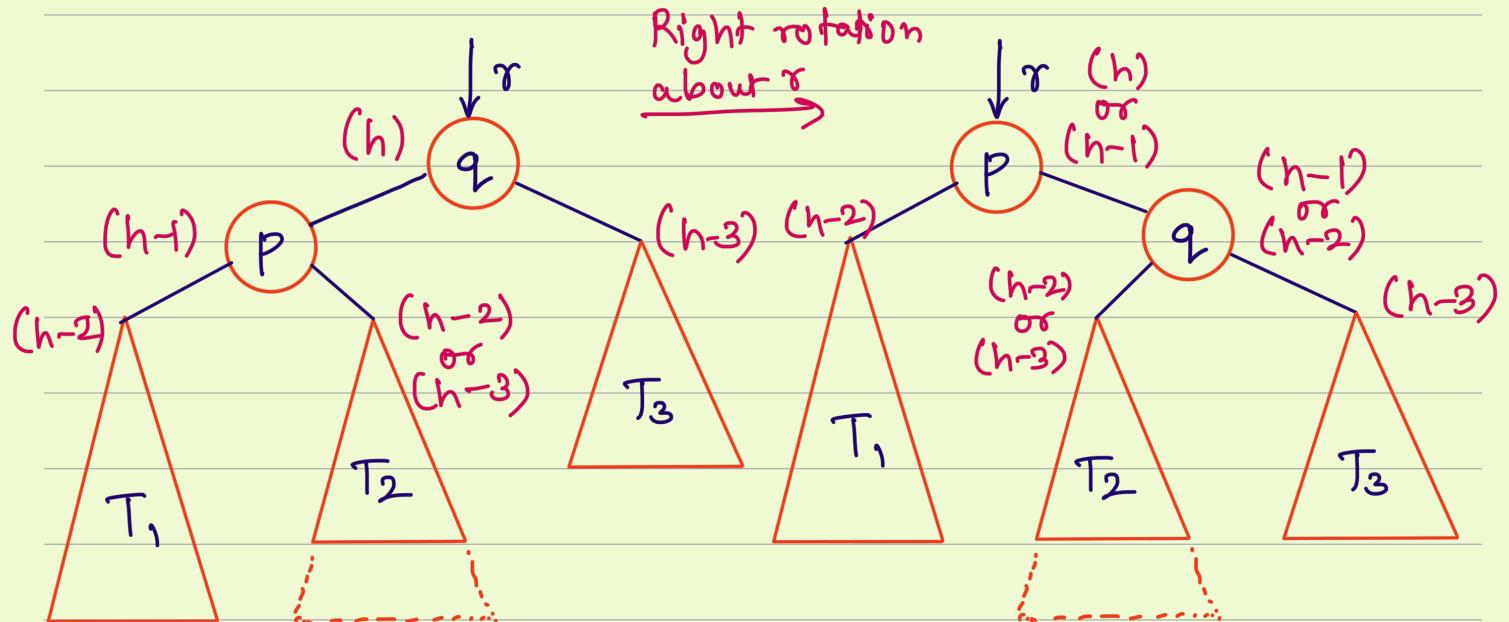
(Right) Rotation

1. Make left subtree of r shorter.
2. Maintain inorder traversal.



inorder (T_1)	P	inorder (T_2)	q	inorder (T_3)
-------------------	-----	-------------------	-----	-------------------

Rebalancing : Case 1.



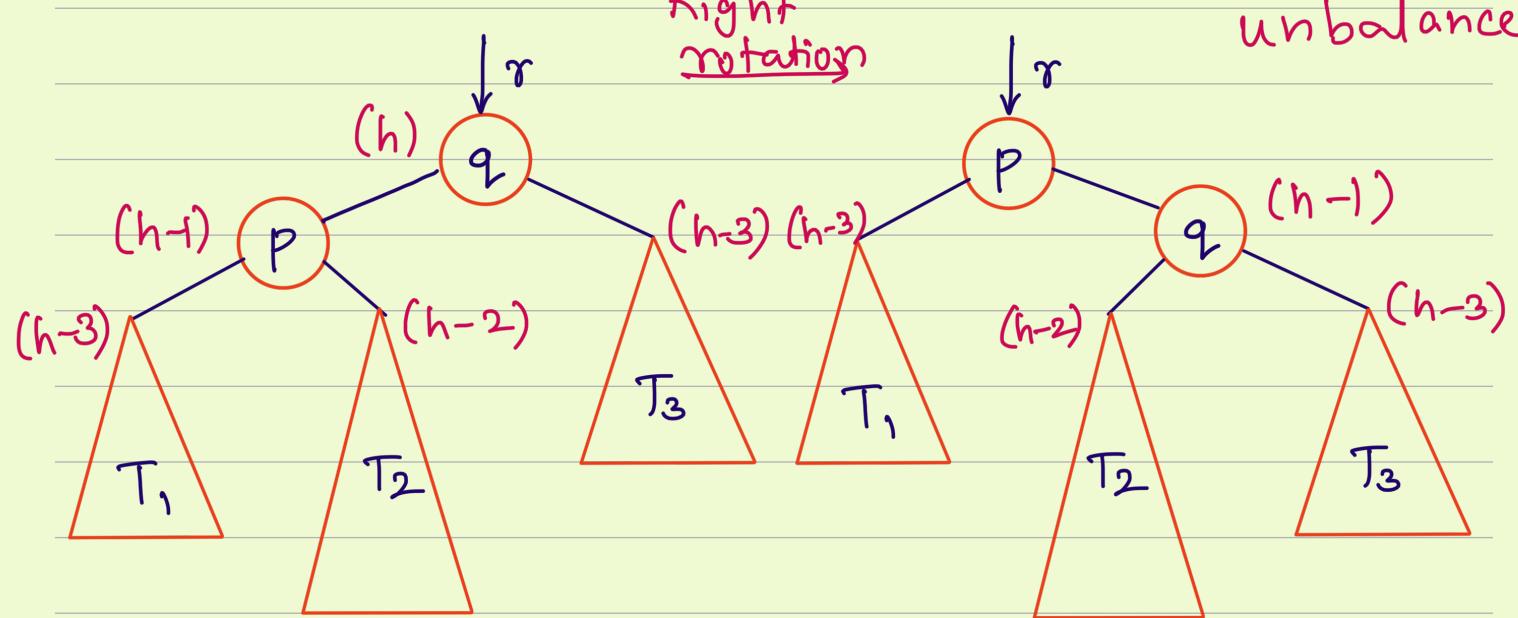
Pre conditions: ① All nodes other than r are balanced.
 ② ht of r 's left subtree = ht of r 's right subtree + 2

Case 1 \equiv height of $T_1 \geq$ height of T_2

Obs: If ht of $T_1 >$ ht of T_2 , then ht of the whole tree ↓.

Rebalancing : Case 2

Still unbalanced!

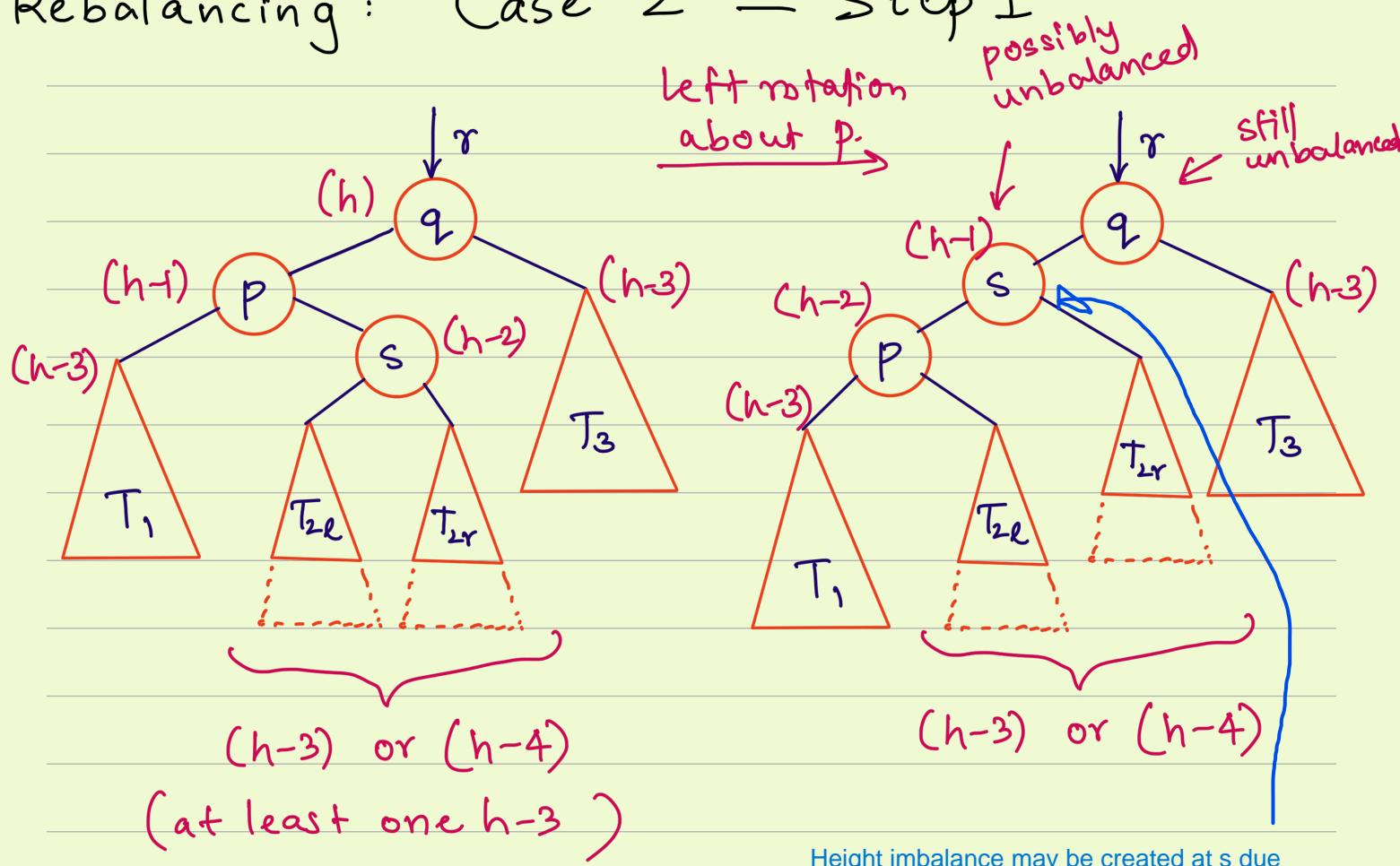


Pre conditions: ① All nodes other than r are balanced.
 ② ht of r 's left subtree = ht of r 's right subtree + 2

Case 2 \equiv height of $T_1 <$ height of T_2 .

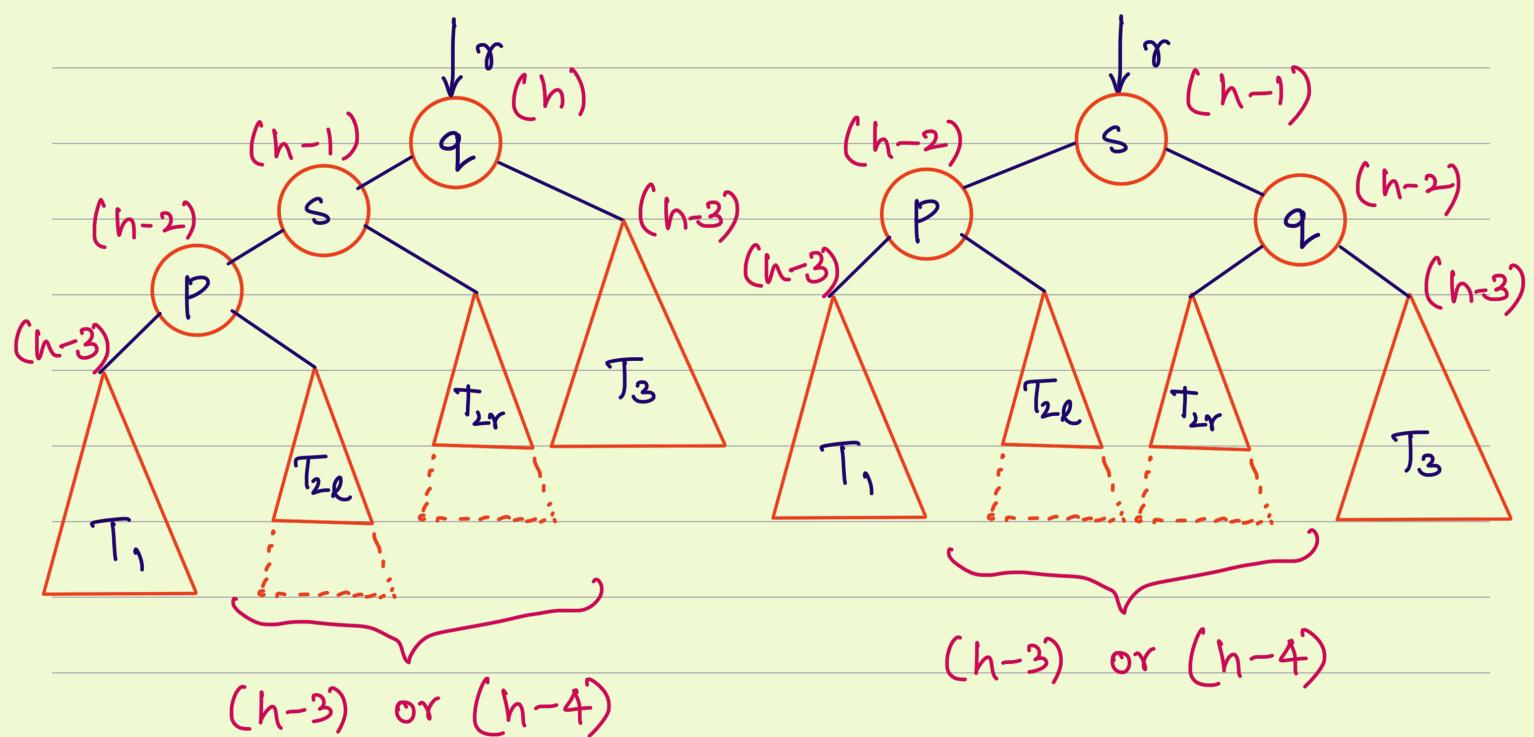
Obs: If ht of $T_1 <$ ht of T_2 , then ht of the whole tree ↓.

Rebalancing : Case 2 — Step 1



Height imbalance may be created at s due to this part $h-2$ and $h-4$

Rebalancing : Case 2 — Step 2



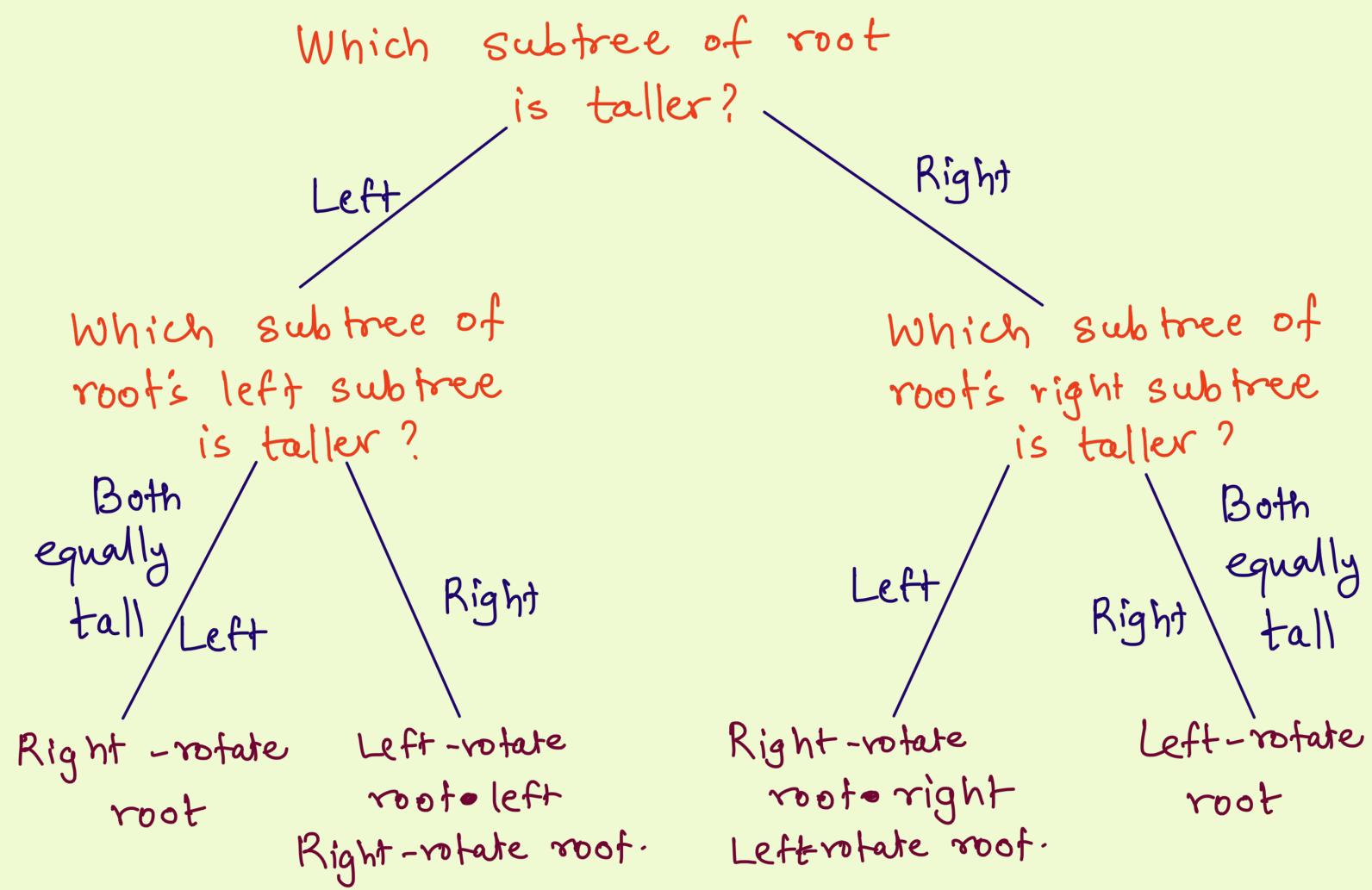
Summary:

Pre conditions:

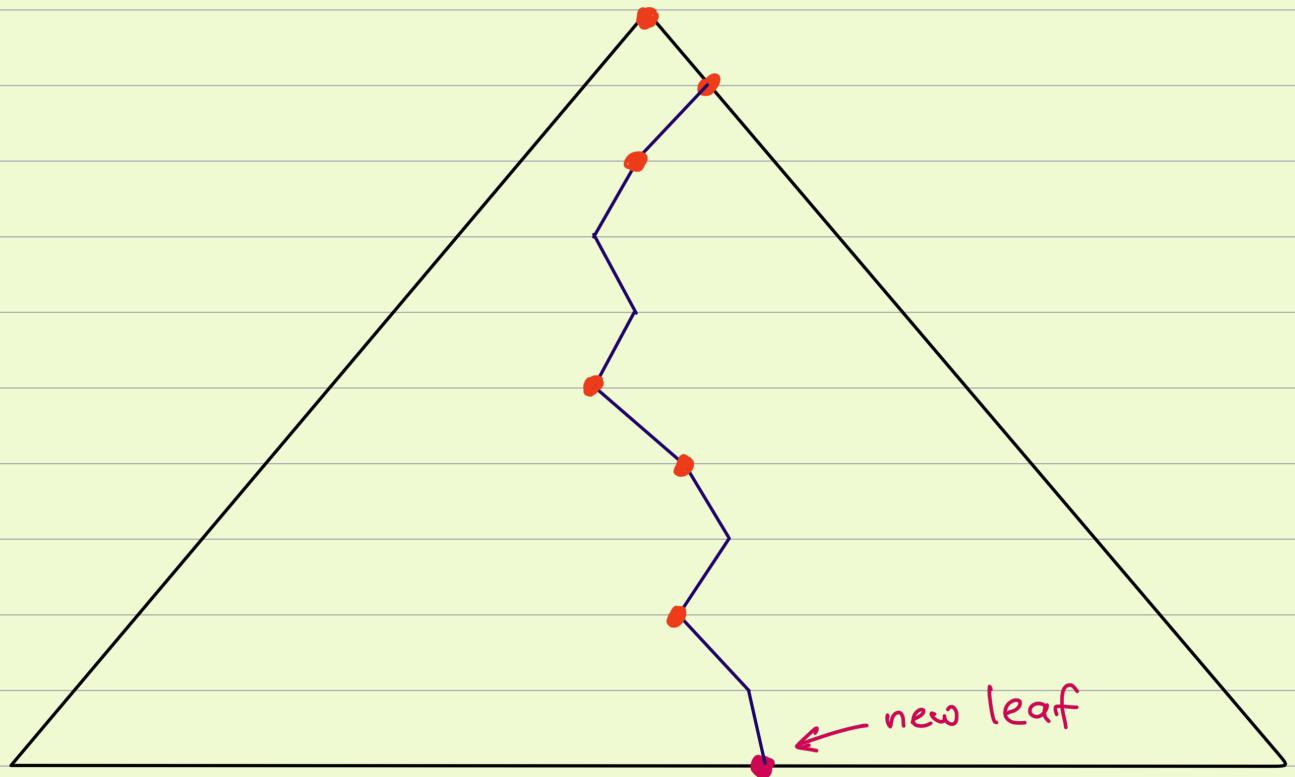
1. All nodes other than root balanced
2. $| \text{ht of root's left subtree} - \text{ht of root's right subtree} | = 2$

Post - conditions

1. All nodes balanced
2. Inorder traversal unchanged



Rebalance after Insertion : the General Case



Obs: A node becomes unbalanced after insertion only if it is an ancestor of the newly created leaf.

This is because we define the height of a tree as $1 + \max(\text{height of subtree1}, \text{height of subtree2})$. The subtrees of only the ancestors of the new node change