Lemma: An AVL tree with height h

has at least F_{h+2} nodes (F_k is Fibonacci number)

$F_0 = 0$ $F_1 = 1$ $F_2 = 1$ $F_3 = 2$ $F_k = F_{k-1} + F_{k-2}$.

$F_{h-2} \geq \phi^{h-2}$ $\boxed{F_{h+2} \geq \phi^{h-2}}$ ϕ = Golden Ratio $= \frac{1+\sqrt{5}}{2} \approx 1.6$

If an AVL tree has n nodes, what is a bound on the height? $\leq \log_\phi n = \Theta(\lg n)$

h

$\log_2 n \leftarrow$ Perfectly balanced

$\lim_{h\to\infty} \frac{F_h}{\phi^h} = \Theta(1)$ $2 > \phi \gtrapprox 1.618$

$F_h \sim \frac{\phi^h}{\sqrt{5}}$

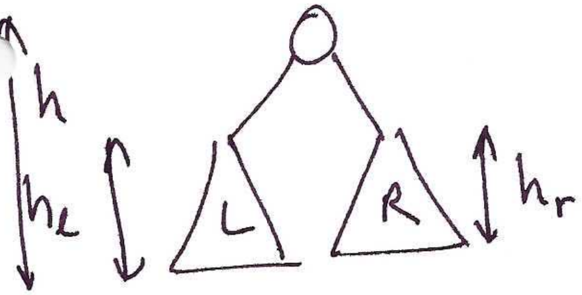
Lemma: An AVL Tree of height ($h \geq 2$) h has at least $F_{h-2} - 1$ nodes

$F_{-2} = 0$ $F_{-1} = 0$ | $F_0 = 0$ $F_1 = 1$ $F_2 = 1$ $F_3 = 2$ - — $F_k = F_{k-1} + F_{k-2}$

Proof: When $h = 2$, the tree has at least 1 node.

$F_{h-2} = 0$. (Base case true)

Induction: Consider tree of height h $(h > 3)$



$$h = \max(h_l, h_r) + 1$$

By induction hypothesis,

$$\#\text{nodes} \geq 1 + (\#\text{nodes in } L) + (\#\text{nodes in } R)$$

$$\geq 1 + F_{h_l} - 1 + F_{h_r} - 1 = F_{h_l} + F_{h_r} - 1$$

Two cases:

Case 1: $h_l = h_r = h-1$. $\#\text{nodes} \geq F_{h-1} + F_{h-1} - 1$

$$\geq F_{h-1} + F_{h-2} - 1 = F_h - 1$$

Case 2: $h_l = h-1$, $h_r = h-2$ (or other way around)

AVL property

$$\#\text{nodes} \geq F_{h-1} + F_{h-2} - 1 = F_h - 1$$

Proof completed. $\blacksquare$

We want to show that #nodes in an AVL tree grows exponentially in h.

#nodes in AVL tree of height h $\geq F_{h-2} - 1$

$$F_h = F_{h-1} + F_{h-2}$$

$$\geq F_{h-2} + F_{h-2} = 2F_{h-2}$$

$$F_h \geq 2F_{h-2} \geq 2^2 F_{h-4}$$

$$\geq 2^3 F_{h-6}$$

By induction, $F_h \geq 2^{h/2} = (\sqrt{2})^h \approx (1.4)^h$

(A more careful analysis of F_h gives ϕ^h.)

If the AVL tree has n nodes and height is h

$$n = \#\text{nodes} \geq (1.4)^h - 1$$

$$h \leq \log_{1.4}(n+1) = \log_2(n+1) \times \boxed{\log_{1.4} 2}$$

$$= O(\log n)$$

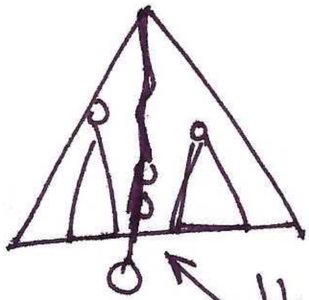
How to maintain AVL property

Insert/delete like a standard BST

Q. When you insert a node in an AVL tree, (height h, nodes n), for how many nodes does the balance factor change?

(R) $\Theta(1)$ ✓ ~~(G)~~ $\Theta(h)$ (B) $\Theta(n)$

$\to \log n$



Heights change ONLY for ancestors.

When a node is inserted, heights change only for ancestors of inserted node.

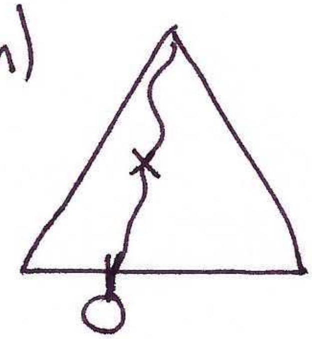
$\Theta(h)$ such nodes $O(\log n)$

Same is true for delete

1) Insert/delete like a standard BST
2) Find all nodes where balance factor is NOT in $\{-1,0,1\}$. (Ancestors of a modified node)
3) For each node, fix the problem.

Q. After an insert/delete, by how much can ~~th~~ balance factor change? → for a single node

✓(R) $\Theta(1)$ (G) $\Theta(\log n)$ (B) $\Theta(n)$



Heights only change by 1.
So balance factor changes by at most 1.

We fix problem bottom-up.

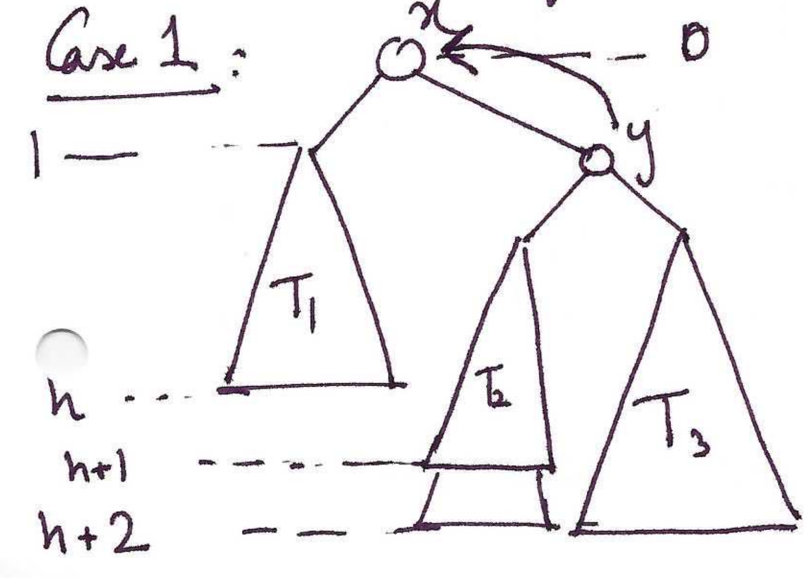
Suppose ~~x~~ balance factor of x is -2.

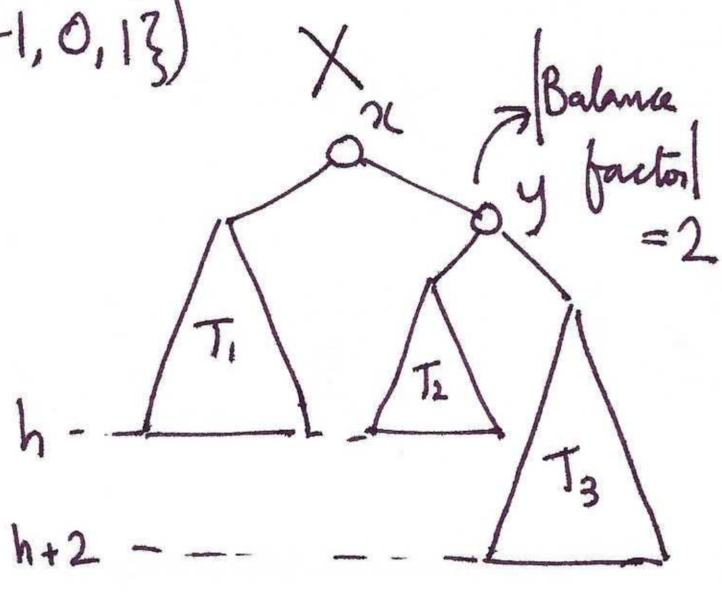
Height of left subtree − Height of right subtree $= -2$

$\boxed{\text{At } x\text{: Height of right} = \text{Height of left} + 2}$

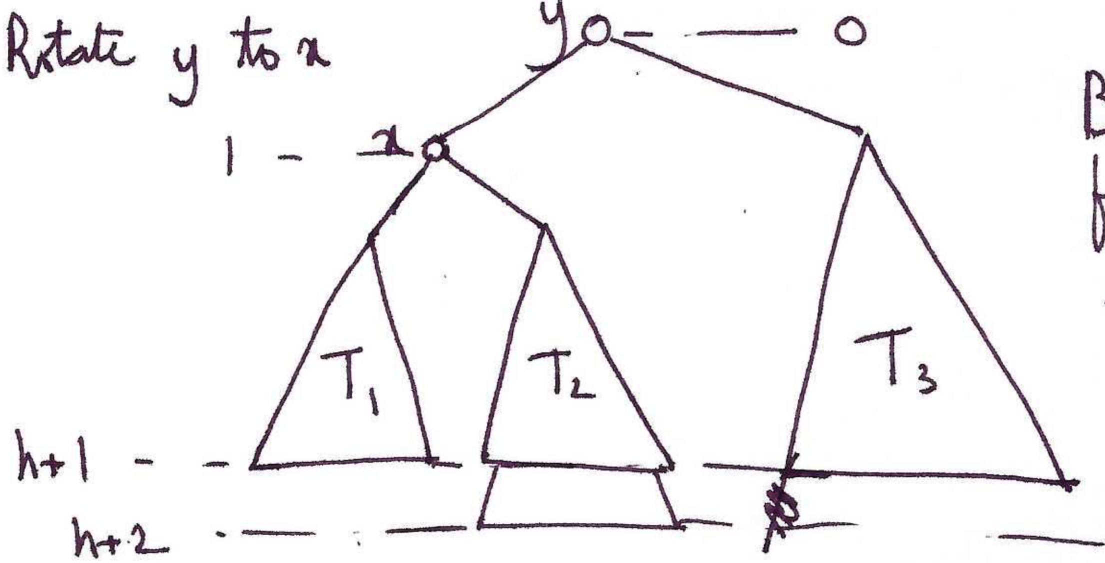
Assume all (strict) descendants of x have a correct balance factor ($\in \{-1, 0, 1\}$)

Case 1:





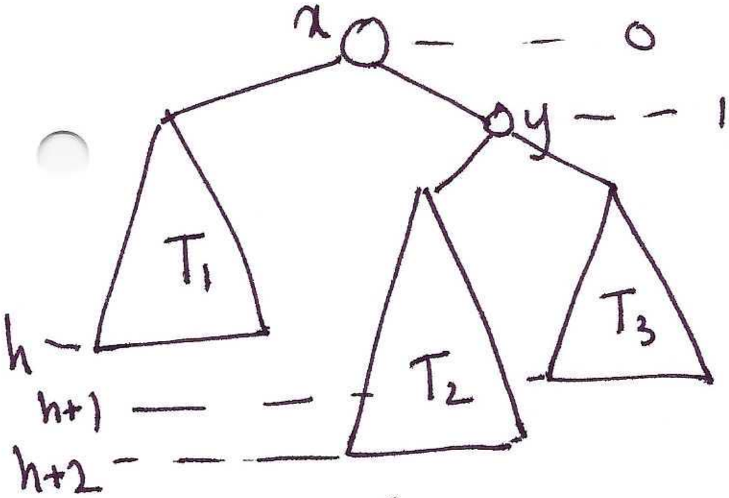
Rotate y to x



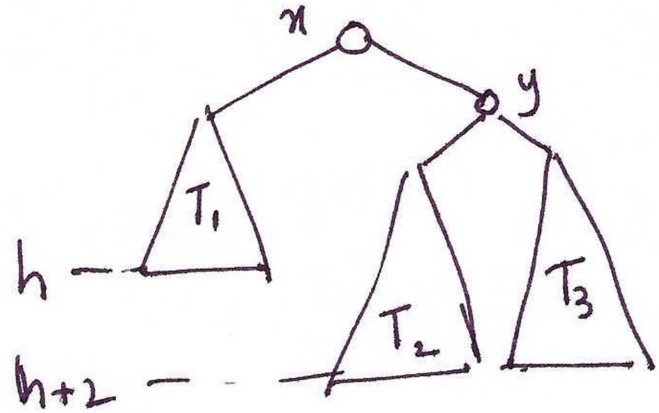
Balance factors are fixed

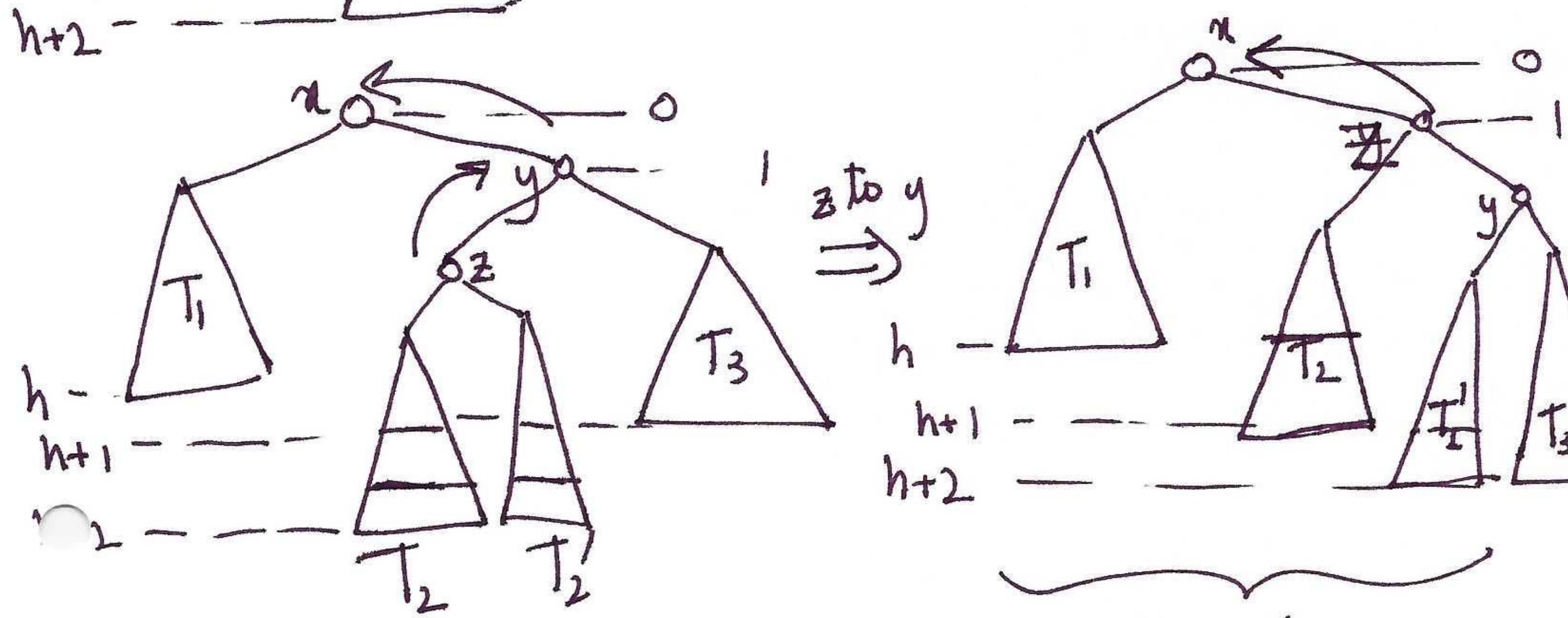
Case 1 : T_3 is the highest

Case 2 : T_2 is the highest



(Done in Case 1)
What about





Case 1