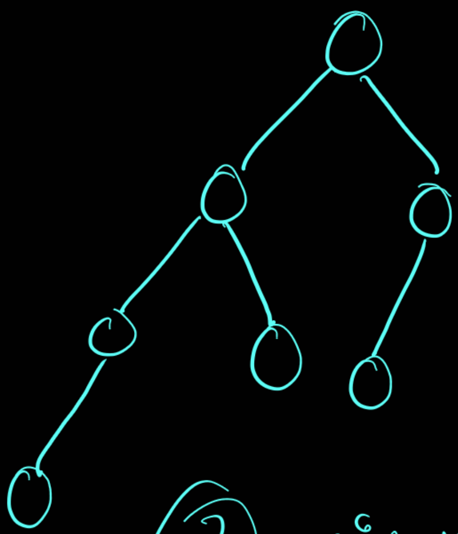
AVL TREES:

→ A BST is said to AVL if

① The max dist. b/w left and right sub-trees of each node is 1.

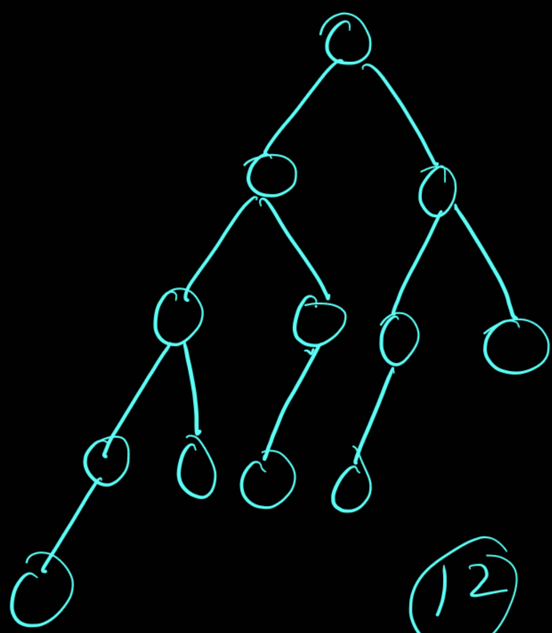
②

Height = 3



7 = min. no. of nodes

Height = 4



12

Let $f(h)$ = min. no. of nodes for height h.

$$\therefore \boxed{f(h) = f(h-1) + f(h-2) + 1}$$

$$f(h-1) = f(h-2) + f(h-3) + 1$$

$$f(h) = f(h-1) + f(h-2) + 1 = 2f(h-2) + f(h-3) + 2$$

$$\therefore f(h) > 2f(h-2) > 2f(h-4)$$

$$\therefore f(h) > 2^2 f(h-4)$$

$$f(h) > 2^3 f(h-6) \Rightarrow f(h) > 2^{h/2} f(0)$$

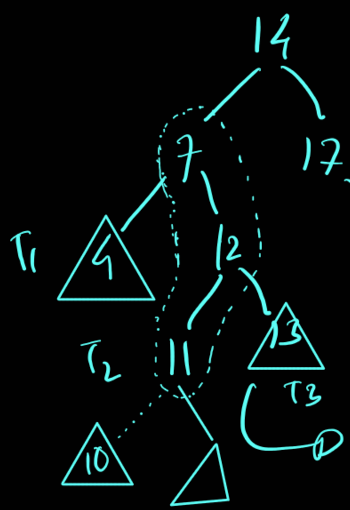
$$\boxed{f(0) = 1}$$

$$\boxed{\therefore f(h) > 2^{h/2}}$$

$$\log f(h) > h/2$$

$$\boxed{\therefore h < 2 \log f(h)}$$

Insert 14, 17, 7, 12, 53, 4, 13, 11, 10



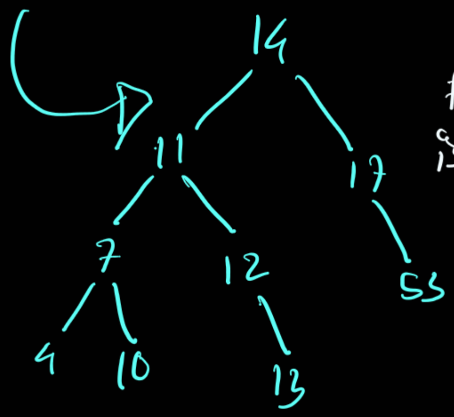
AVL ✓

Now Insert 10

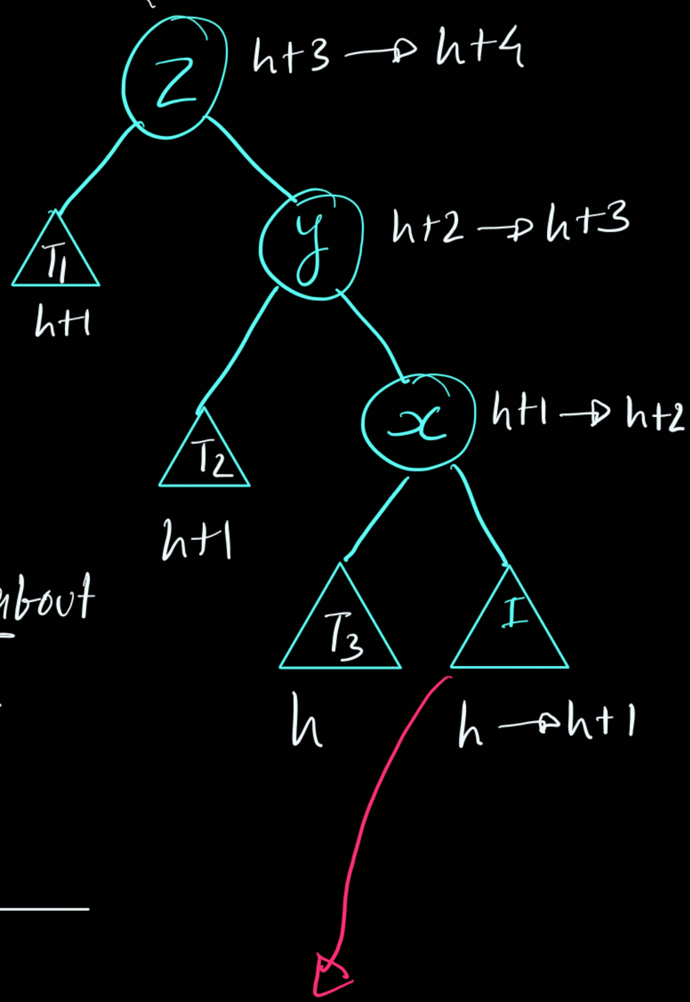
7 12 11 ⇒ 7 < 11 < 12

T1 T2 T3 T4

4, 10, NULL, 13



first node, which is not AVL balanced



Insertion was made here

Initially, the subtree I has height = $\boxed{h}$

Now height of T_3 can be:

- $\hookrightarrow h+1$ ✗
- $\hookrightarrow h-1$ ✗
- $\hookrightarrow h$ ✓

Think about this

Now, the height of y can be:

- $\hookrightarrow h+2$ ✓
- $\hookrightarrow h+3$ ✗
- $\hookrightarrow h+1$ ✗

Thus, the height of $\boxed{T_2 = h+1}$

(same logic as T_3)

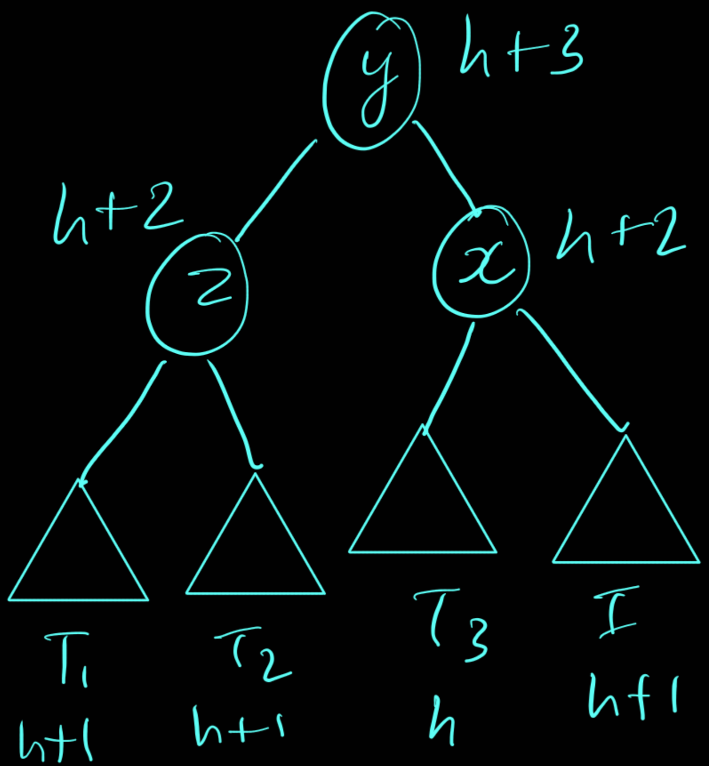
Now, the height of z
can be:

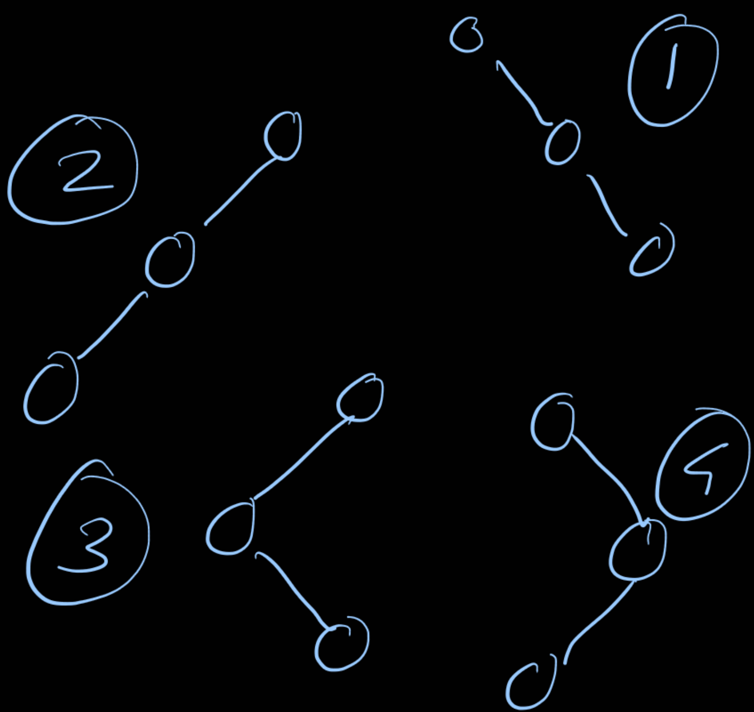
↳ h+3 ✓
↳ h+4 ✗
↳ h+2 ✗

Thus, the height of $\boxed{T_1 = h+1}$

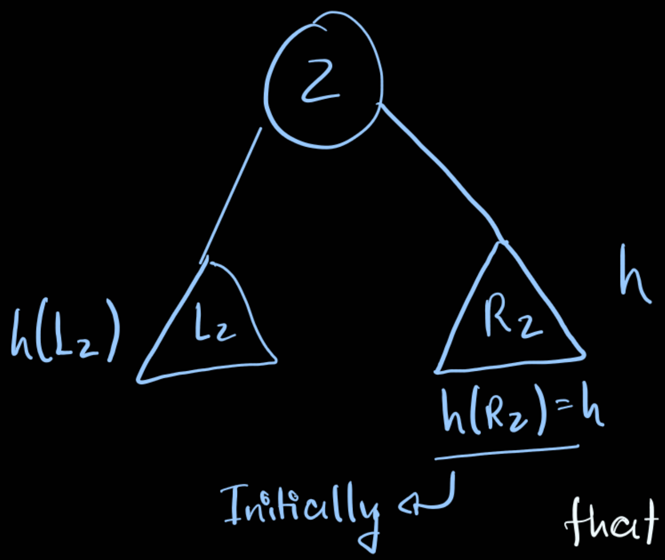
(As z becomes unbalanced)

Now, rotate the subtree in ACW dir^n





Thus, in the above case,



If the height of R_z is h, then the height of L_z must be $h-1$.

Reason: Z is the node that gets disbalanced after $height(R_z) \rightarrow h+1$

$\therefore diff(L_z \,\&\, R_z) = 2$ (diff ↑es)

$\because h(L_z) < h(R_z)$

& $diff = 2$ (After $h(R_z)$ ↑es)

[Let $h(L_z) = h'$] $\therefore (h(R_z)+1) - h(L_z) = 2$

$\therefore h + \cancel{1} - h' = \cancel{2}\,1$

$\therefore \boxed{h' = h-1}$

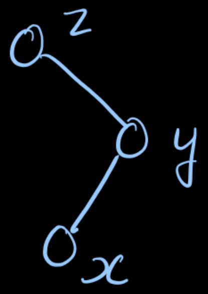
Similarly, for node x and y,

$$\boxed{h_L = h_R}$$

→ If $h_L < h_R \Rightarrow$ node will get disbalanced after h_R ↑es

→ If $h_L > h_R \Rightarrow$ height of the parent node won't increase.

Case 2:

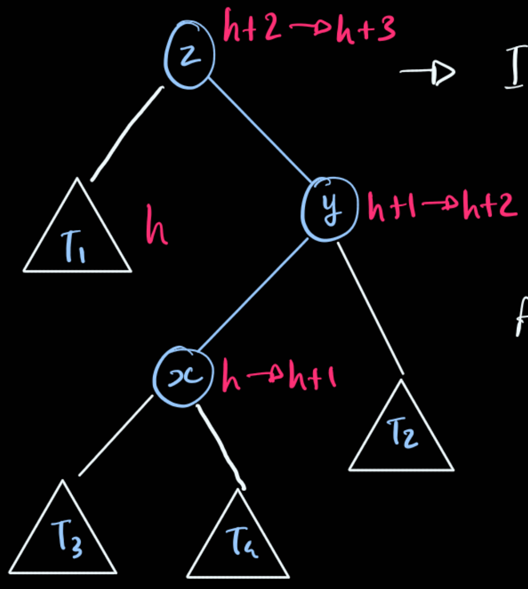


$z \rightarrow$ smallest value

$y \rightarrow$ largest value

$x \rightarrow$ middle value

$z \rightarrow$ the node which gets disbalanced

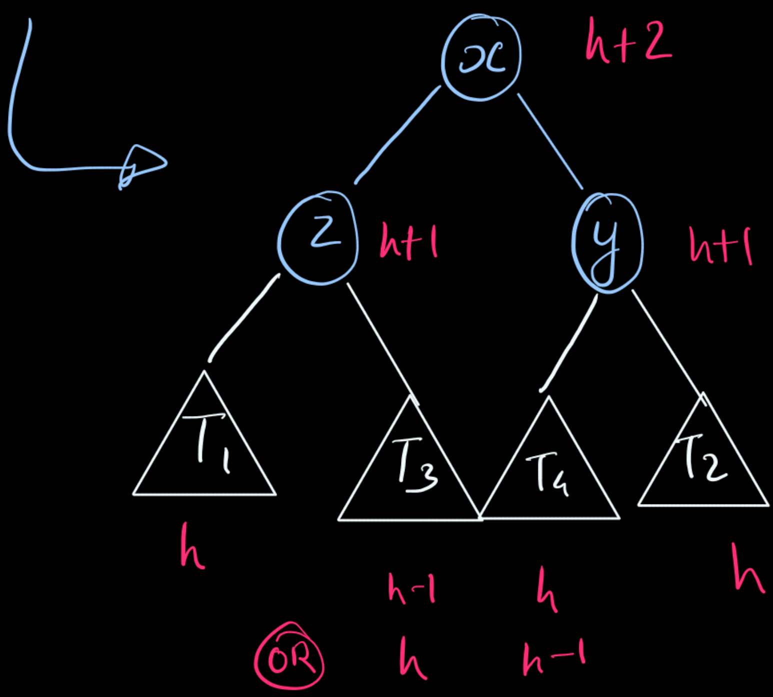


→ Initially, let the height of x be h.

$\therefore$ $\boxed{h(T_2) = h}$ (As h_y increases without getting disbalanced)

Also,

$\boxed{h(T_1) = h}$



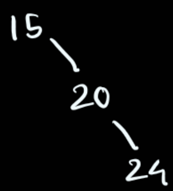
Eg: 15, 20, 24, 10, 13, 7, 30, 36, 25

Insert this

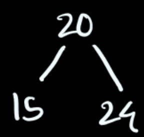
1 → 15

2 →

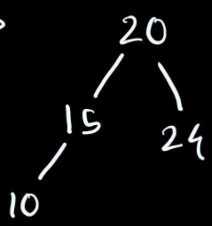
3 →



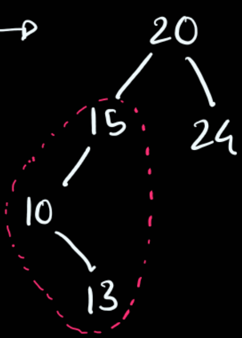
NOT AVL ⇒



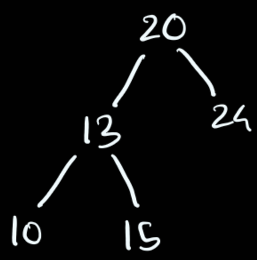
4 →



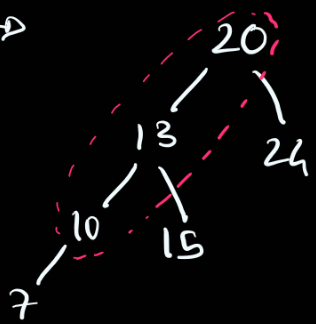
5 →



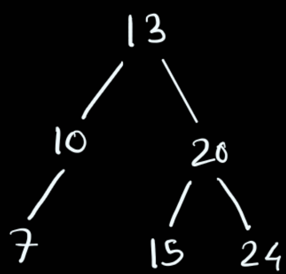
NOT AVL ⇒



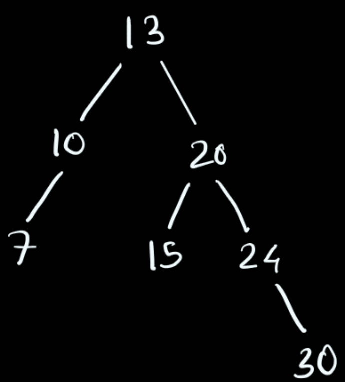
6 →



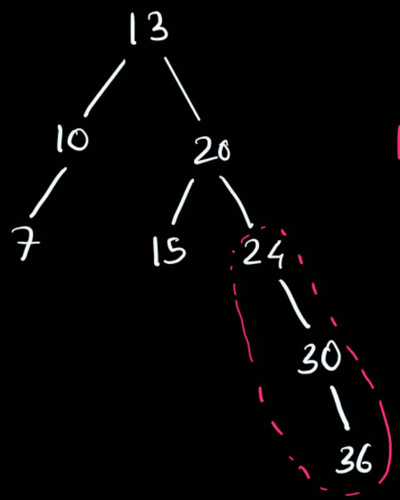
NOT AVL ⇒



7 →

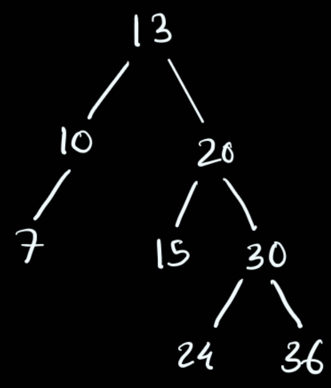


8 →

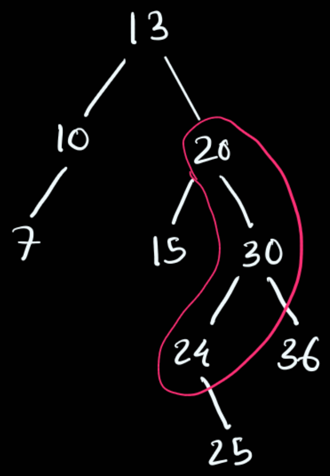


NOT AVL

⇒

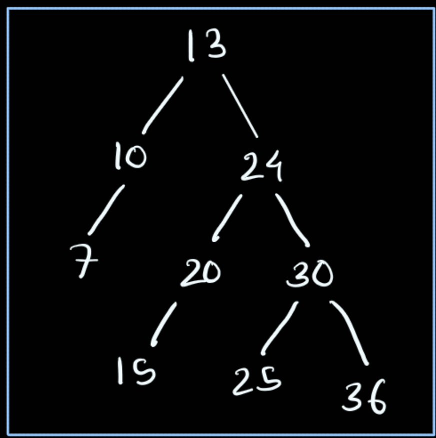


9 →



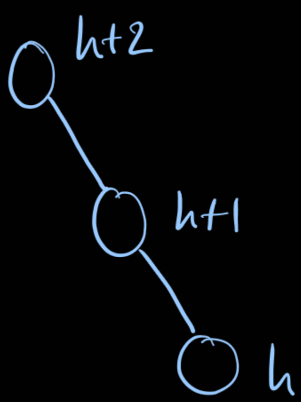
NOT AVL

⇒

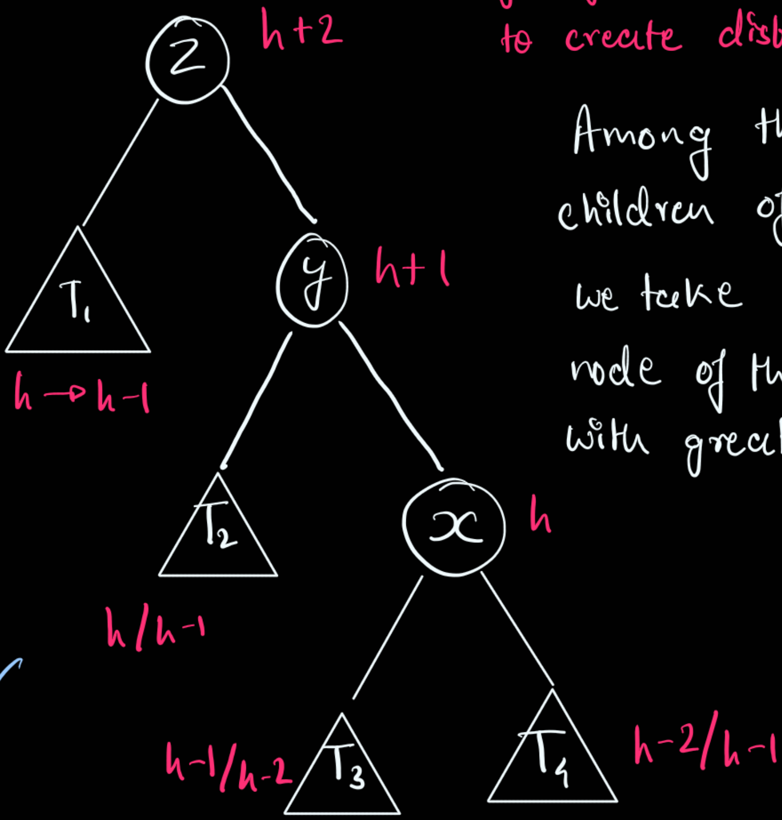


Deletion :

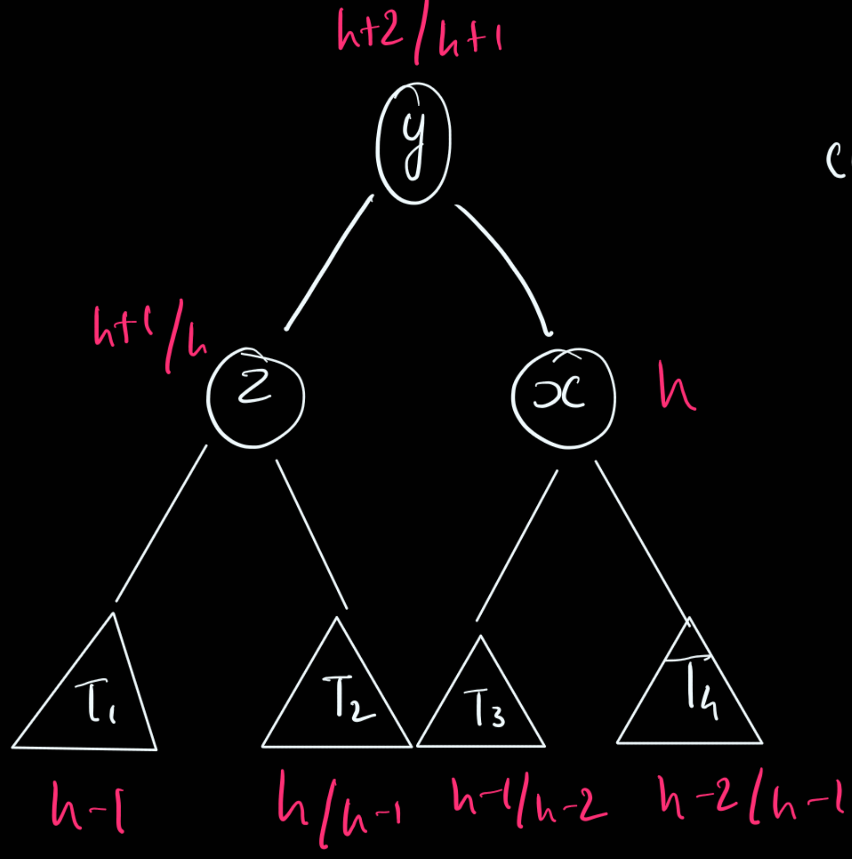
Case (i) :



height(y) must be h+1 to create disbalance

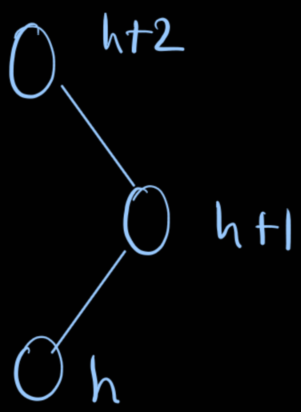


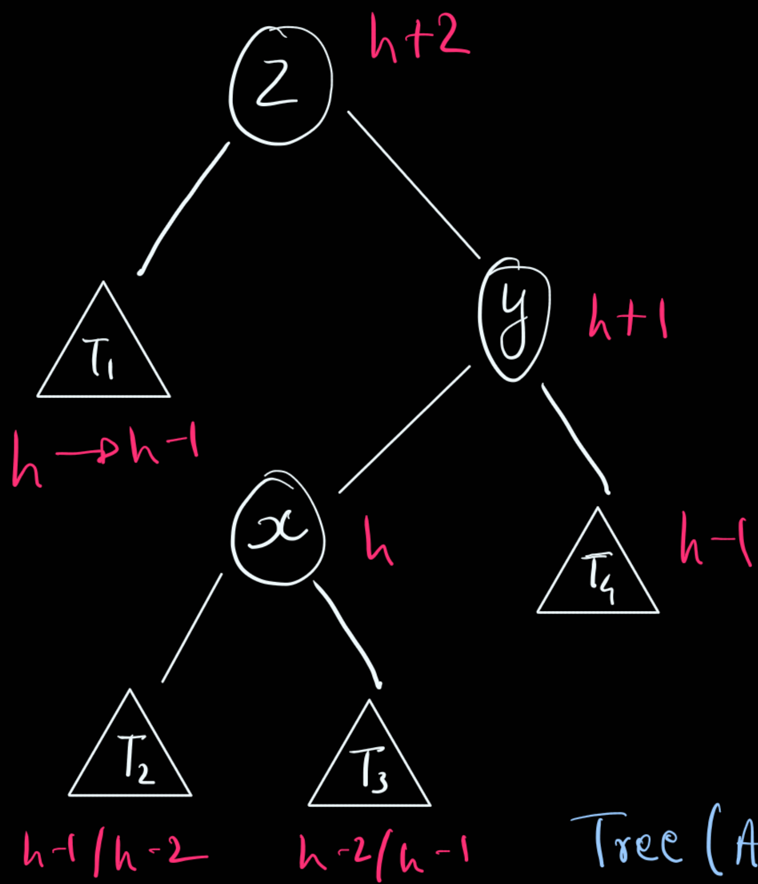
Among the children of y, we take x as the node of the subtree with greater height.



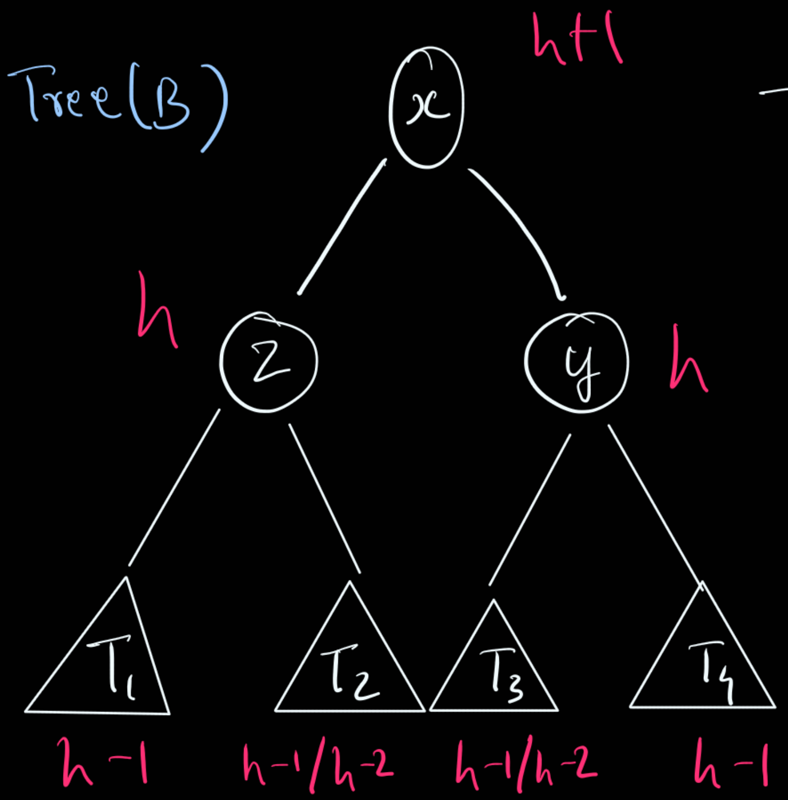
If h(y) = h+1, continue this until the tree is balanced.

Case ②:





Tree (A)



→ $h(x) = h+1$

∴ the height of this subtree [x] decreased by 1

Possibly, disbalancing its parent node.

Hence, Continue this untill the tree is balanced.

In the Tree(A), if $h(x) = h(T_4)$, then take T_4 as x and follow the st. line case.

Why case ① is better than case ②.
