Binary Search Tree : Need not be balanced
(BST)

- Balance Factor:

$$BF(n) = \text{Height of right subtree} - \text{Height of left subtree}$$

$BF(n) > 0$: right heavy

$BF(n) < 0$: left heavy

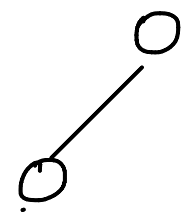
- AVL Tree which are BST, $|BF(n)| \leq 1$

Adelson, Velsky, Landis

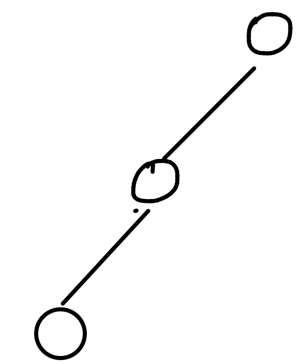
- Property $h = O(\log(n)) \leq C \log(n)$ (Prove it later)

Basic Imbalance

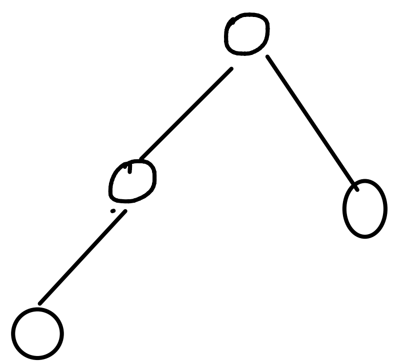
Balanced

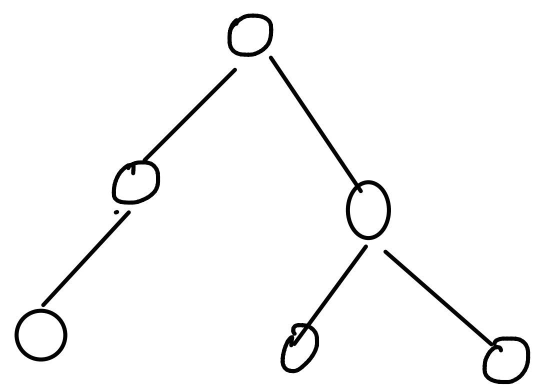


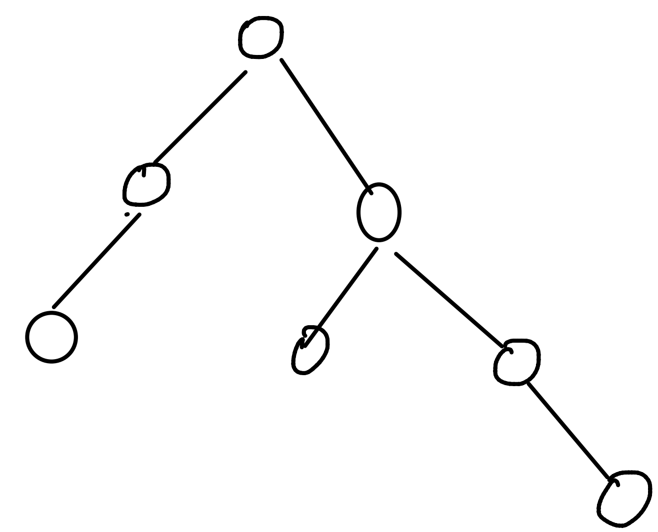
Un Balanced

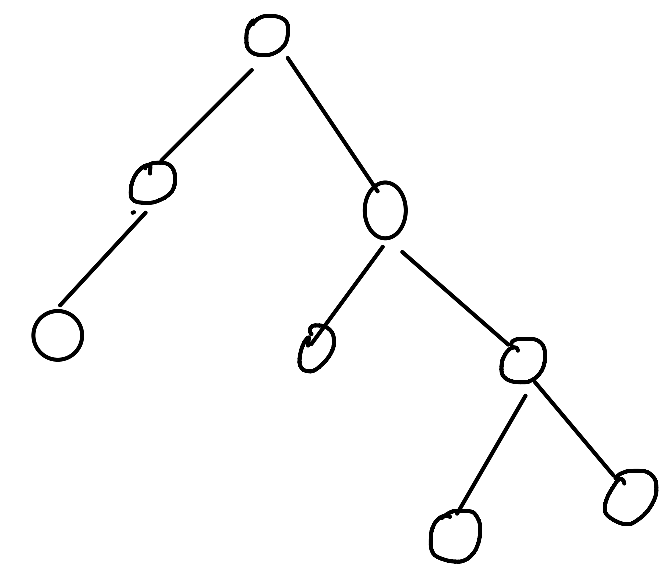


Balanced

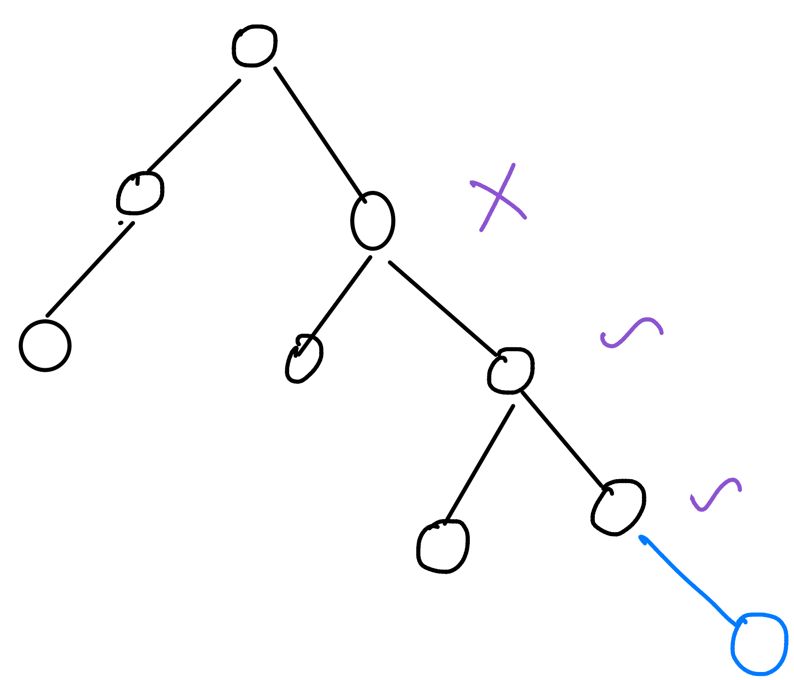








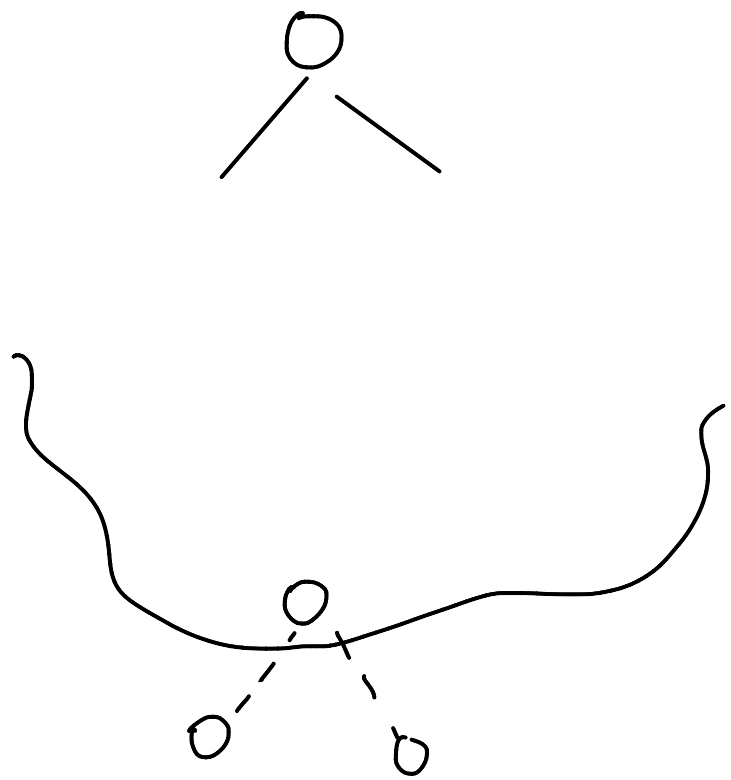
Focus on level at which balance is lost



Insertion

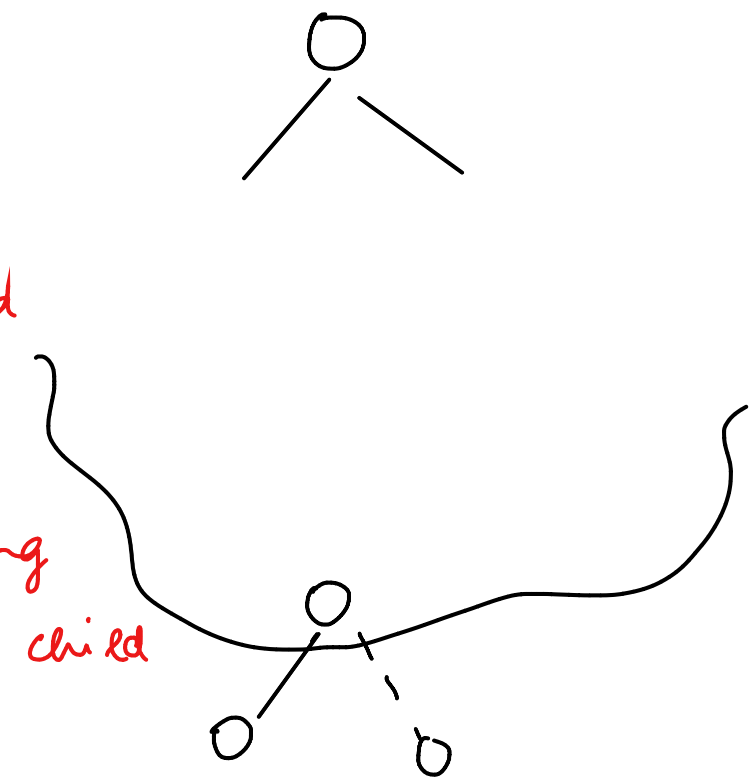
Case 1

Insert to either left or right of leaf



Case 2:

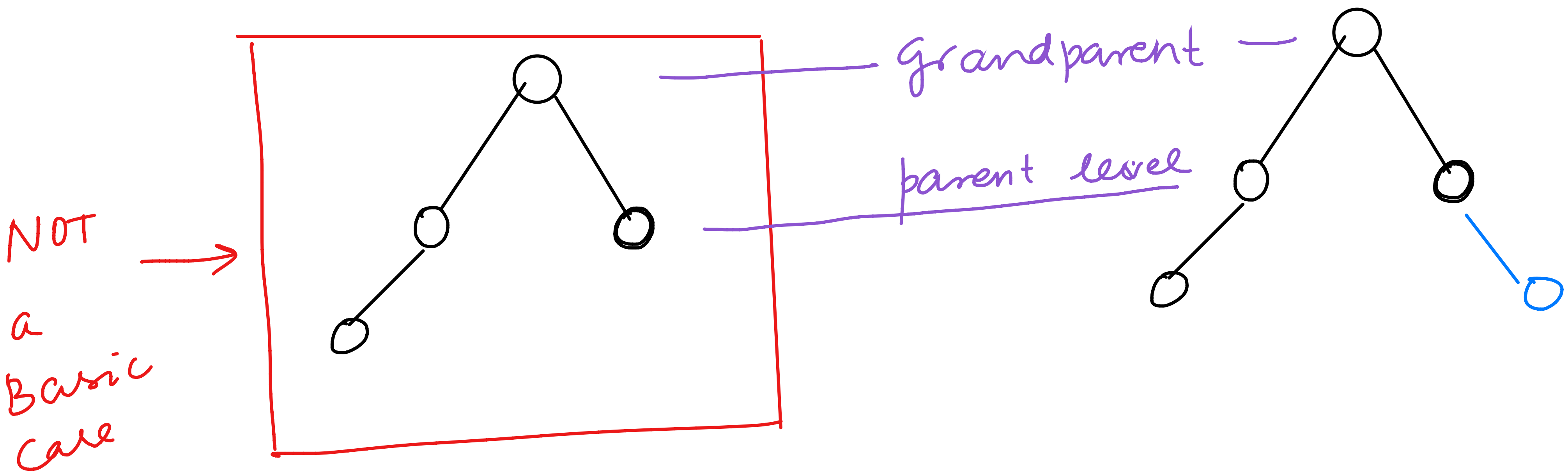
already one child is there and we are adding a second child



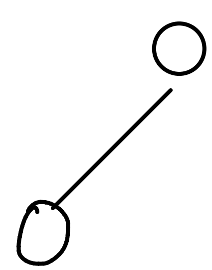
Immediate parent to which the new node is inserted will not have imbalance

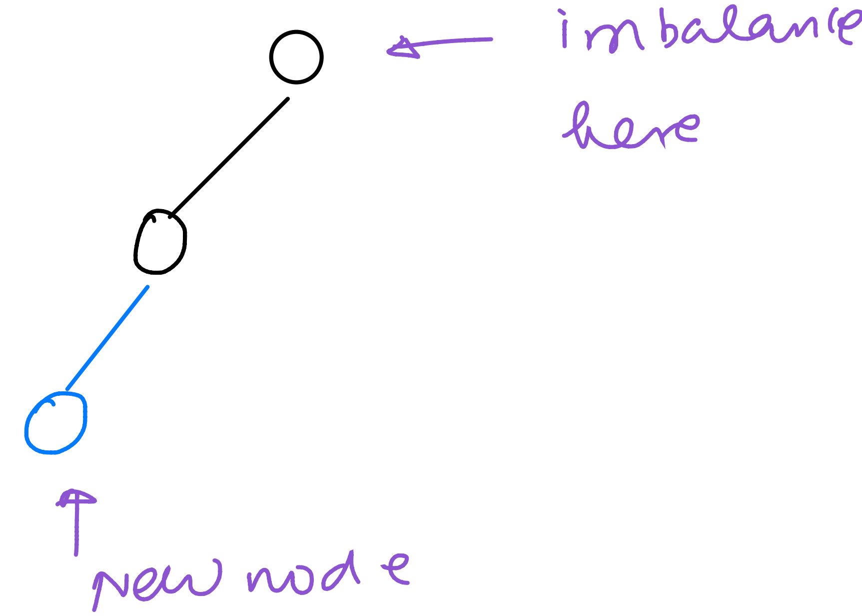
Basic Case : Simplest Configuration in which Adding New node causes imbalance

Before adding node no imbalance



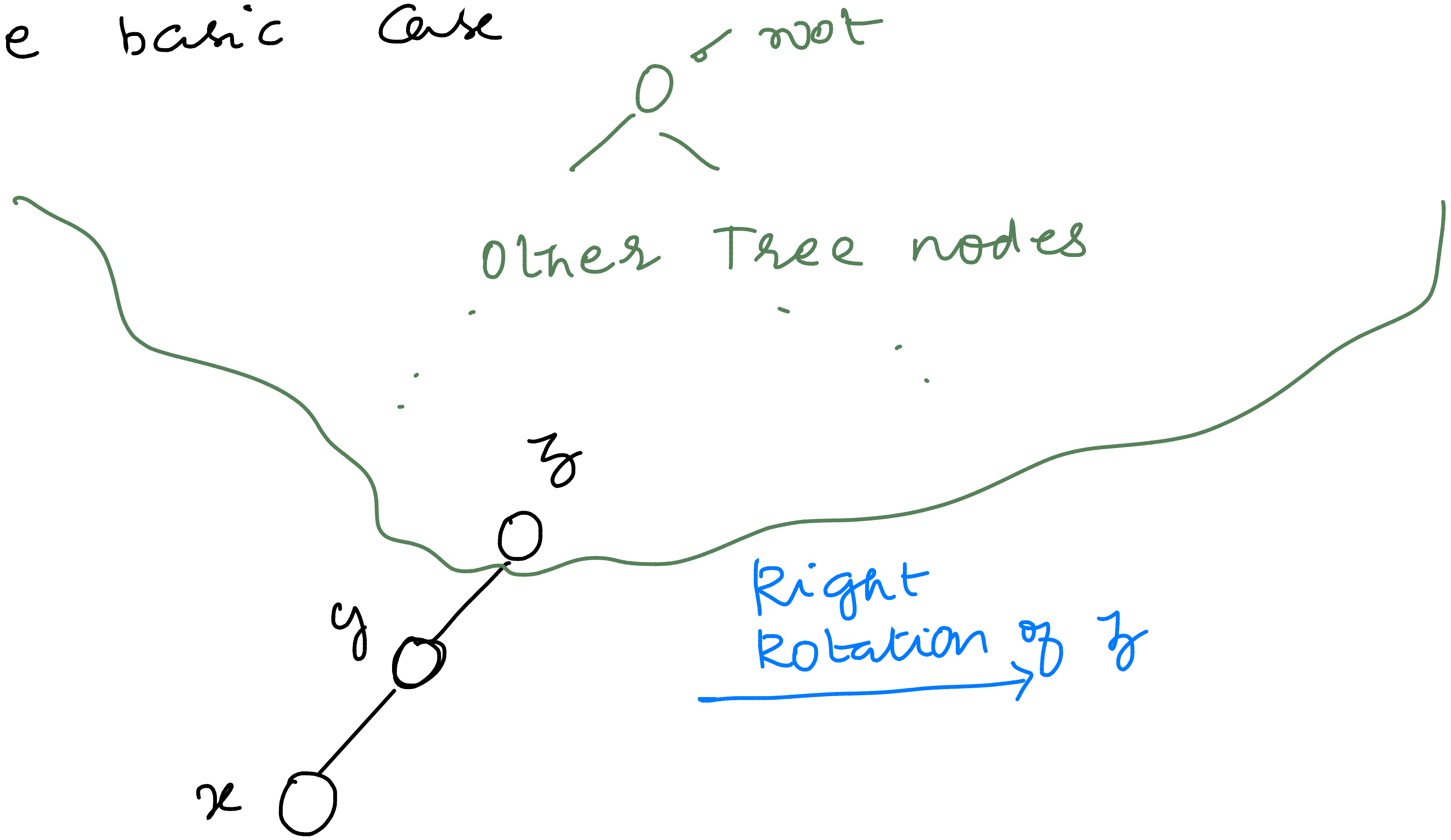
Basic Case : Immediate Grandparent has imbalance





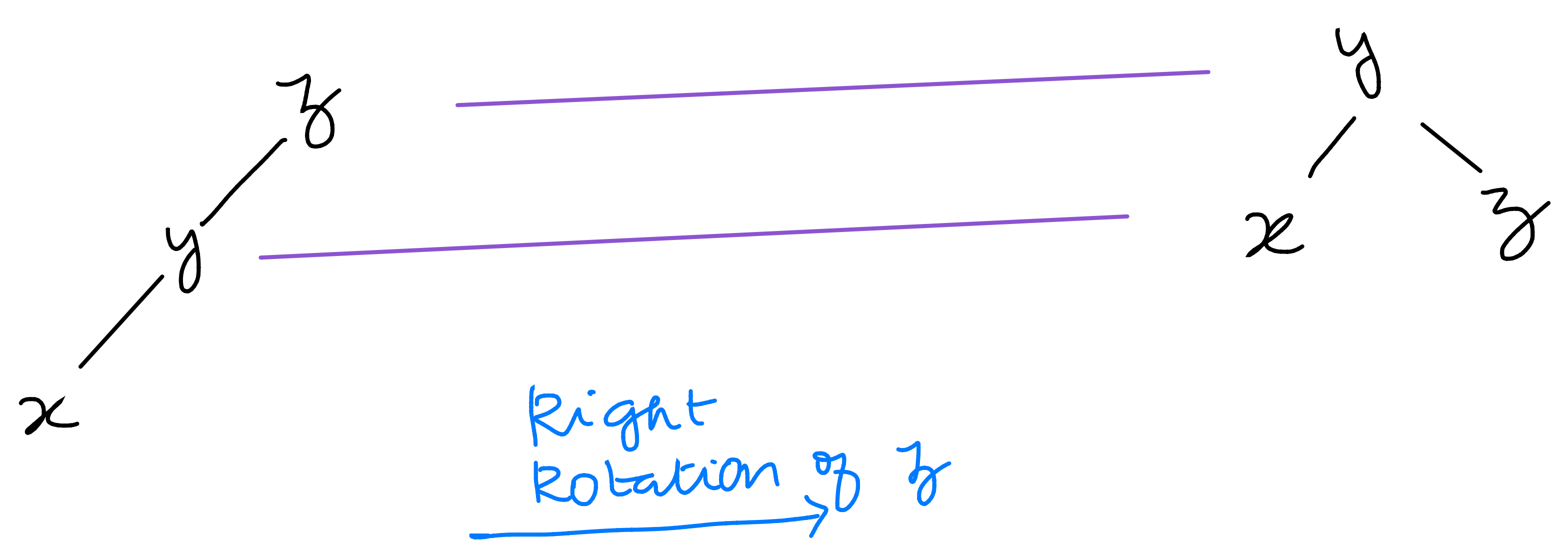
Resolving the basic case

Left Left Imbalance



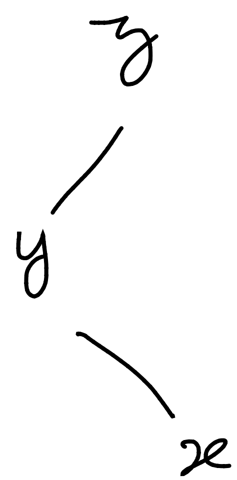
Remove rest of the tree from our mind and focus only on node where imbalance has occured

Left Left Imbalance

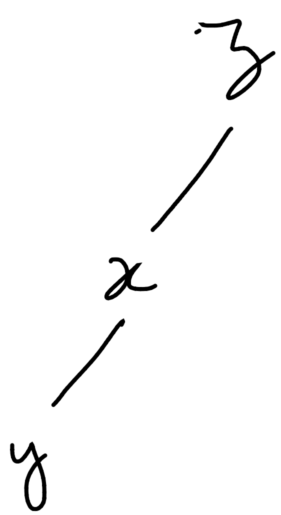


Left Right

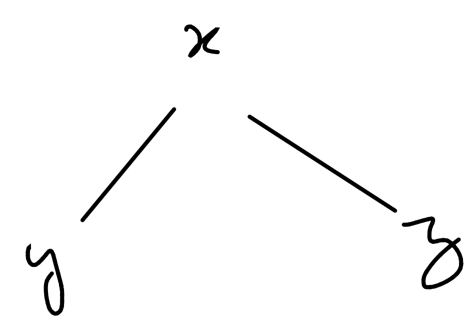
Imbalance



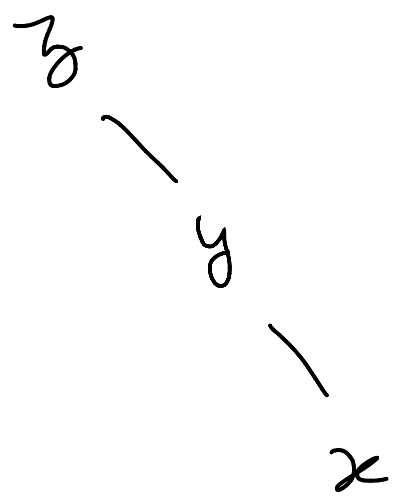
First reduce it to left left imbalance by doing left rotation of y



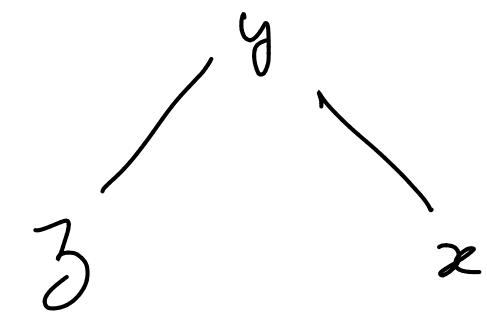
Right Rotation of (z)



Right Right Imbalance

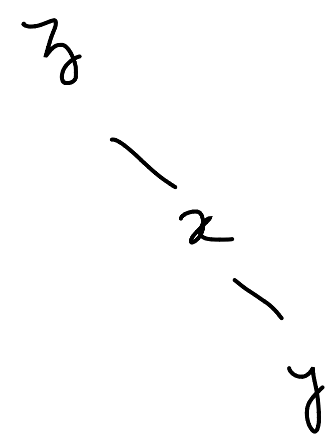


Rotate z to left →

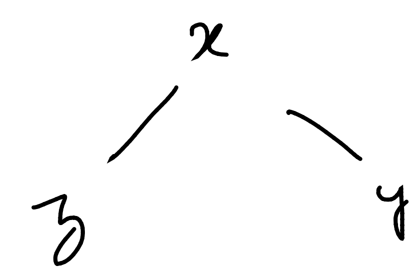


Right Left Imbalance

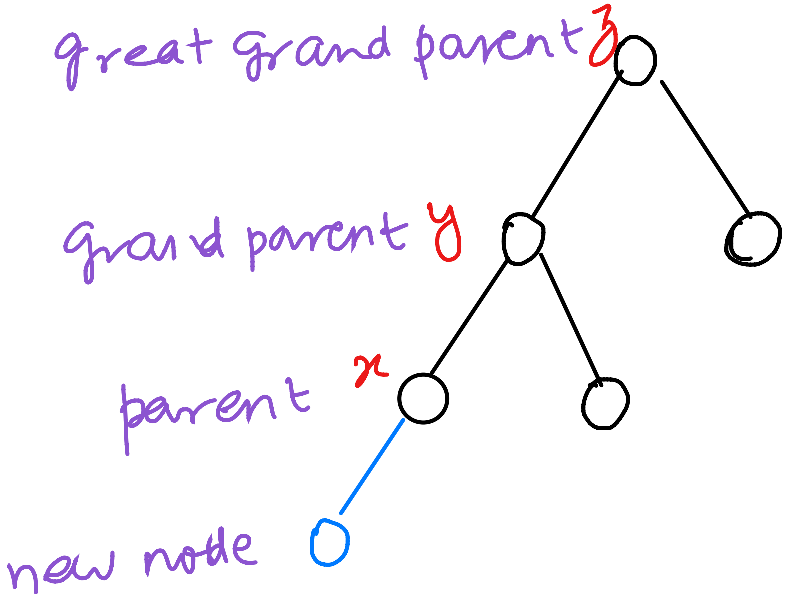
Rotate y to right →



Rotate z to left →

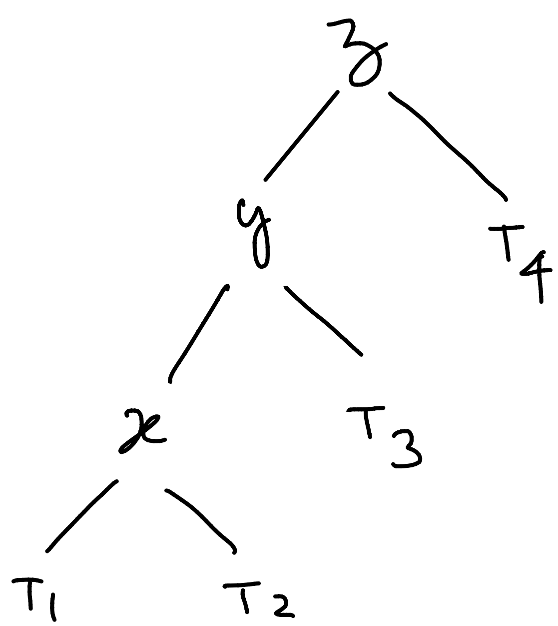


General Case of Imbalance



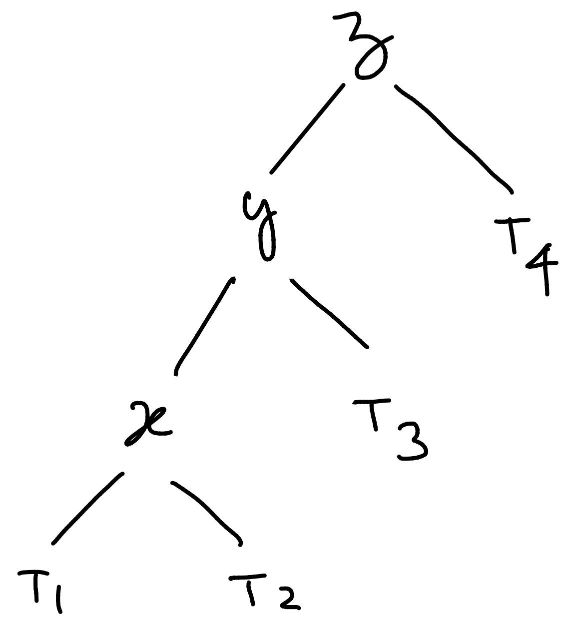
xyz shifted up one level

General case: xyz shifts up by arbitrary levels

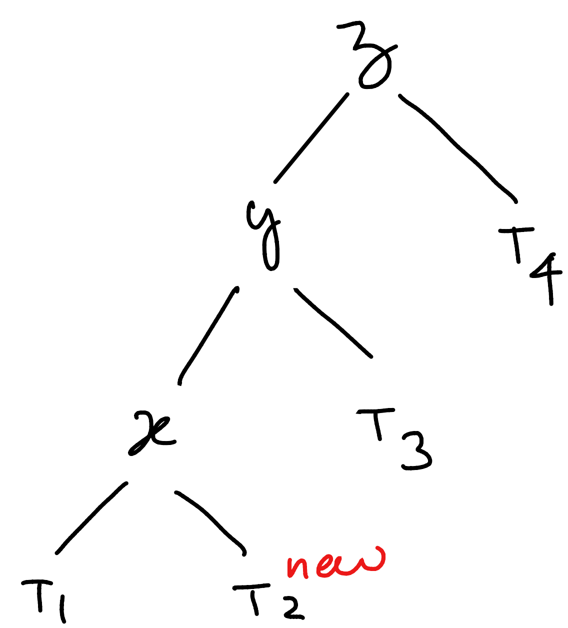


Left Left Imbalance

(Property I) Before addition all nodes are balanced



(Property II) After addition nodes x, y are balanced and z is imbalanced



T_2^{new} is T_2 after adding New Node

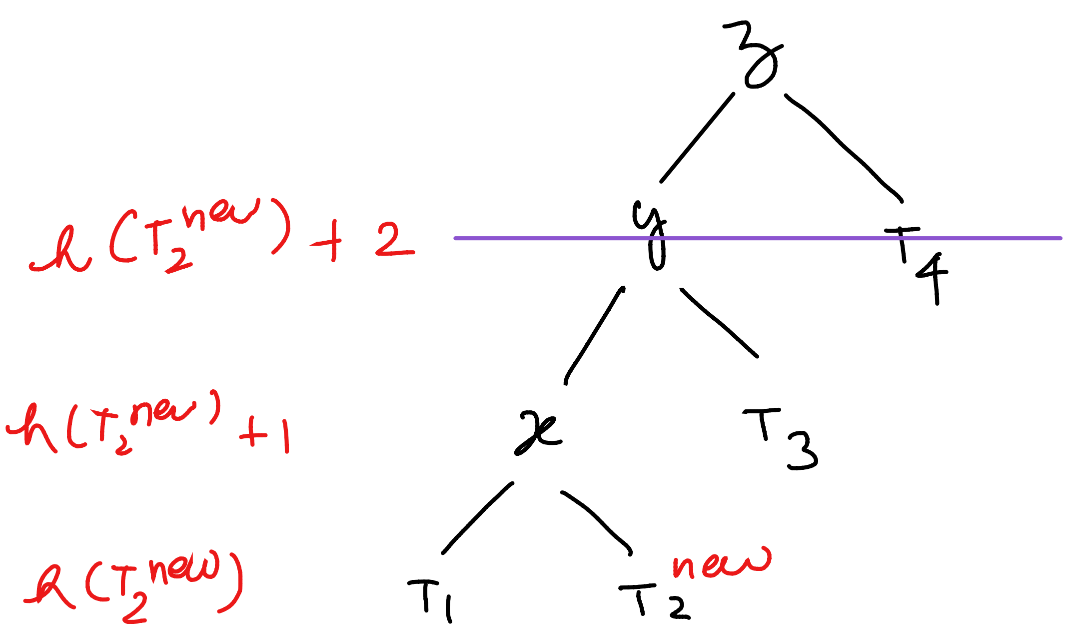
Property I $\Rightarrow$

- ✗ $h(T_2) = h(T_1) - 1$ (not possible; cannot change height at level x by adding new node to T_2)
- ✗ $h(T_2) = h(T_1) + 1$ (not possible; because after adding new node to T_2, level x itself becomes unbalanced $\Rightarrow$ contradicts property II)
- ✓ $h(T_2) = h(T_1)$

$$\Rightarrow \quad h(T_2^{new}) = h(T_1) + 1$$

Property II

After adding, imbalance is at level z; T_4 is lighter



$$h(T_2^{new}) + \cancel{2} - h(T_4) = \cancel{2}$$

$$h(T_2^{new}) = h(T_4)$$