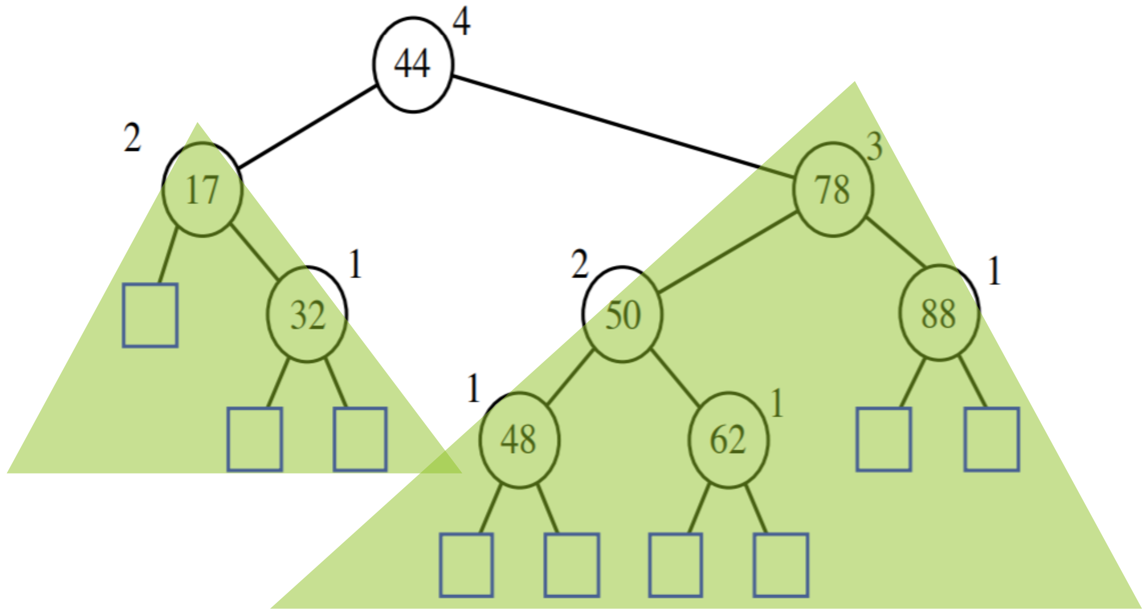
CSI 2110 Tutorial (Section A)

Yiheng Zhao
yzhao137@uottawa.ca
Office Hour: Fri 13:00-14:00
Place: STE 5000G

Review AVL Tree

A **binary search tree** such that for every internal node v, the **heights** of its **left subtree** and **right subtree** only differ by **at most 1**



The height includes the dummy nodes

Review AVL Tree

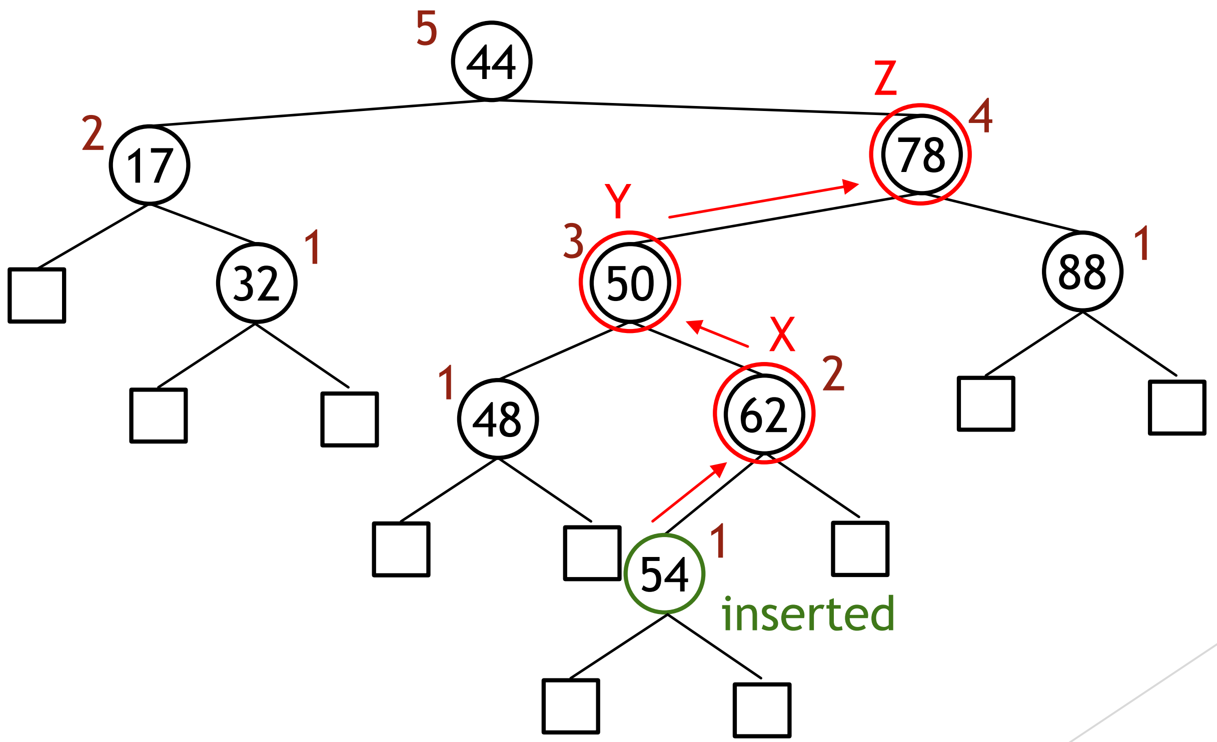
Insertion

1) Insert new node according to rule of binary search tree

2) Rebalancing

step 1: trace back to find the node whose **grandparent** is the **first unbalanced node**.

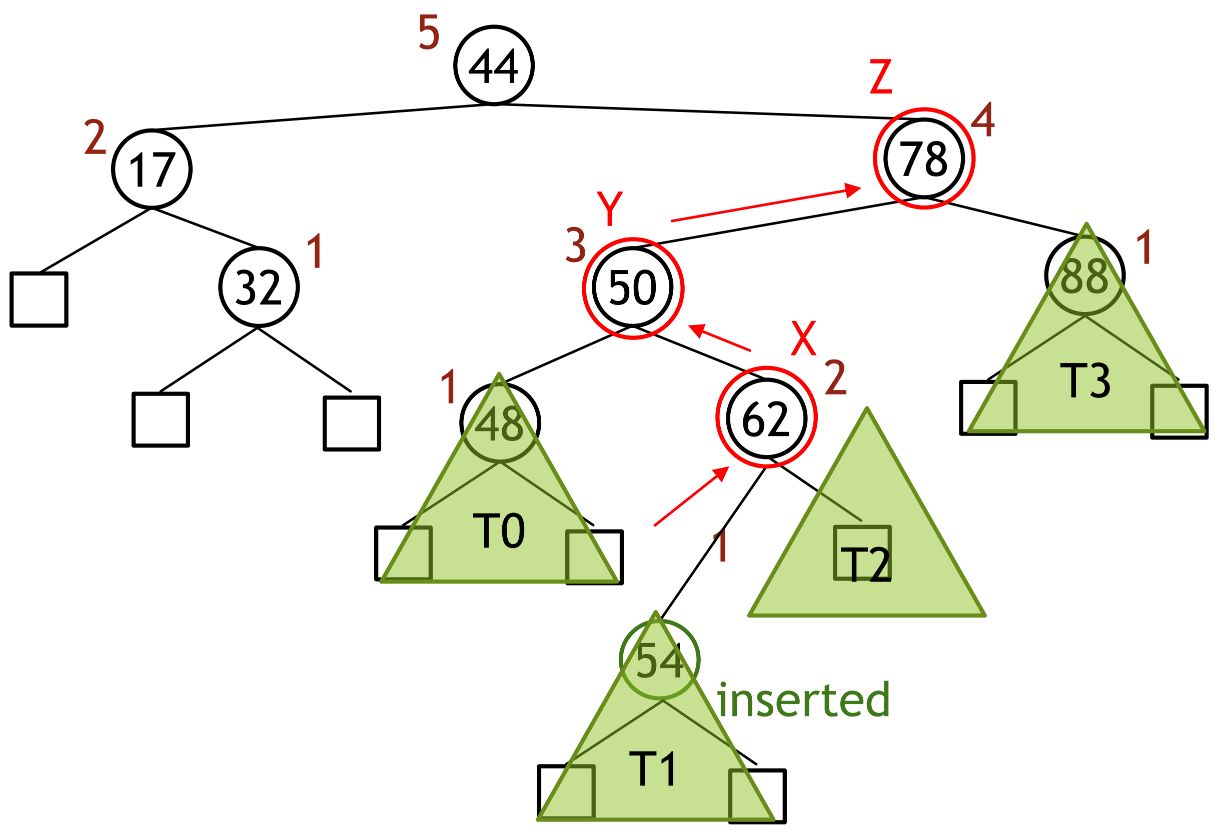
label the node (x), parent (y), and grandparent (z).



Review AVL Tree

2) Rebalancing

step 2: label the subtrees from **left to right** as T0, T1, T2, T3.

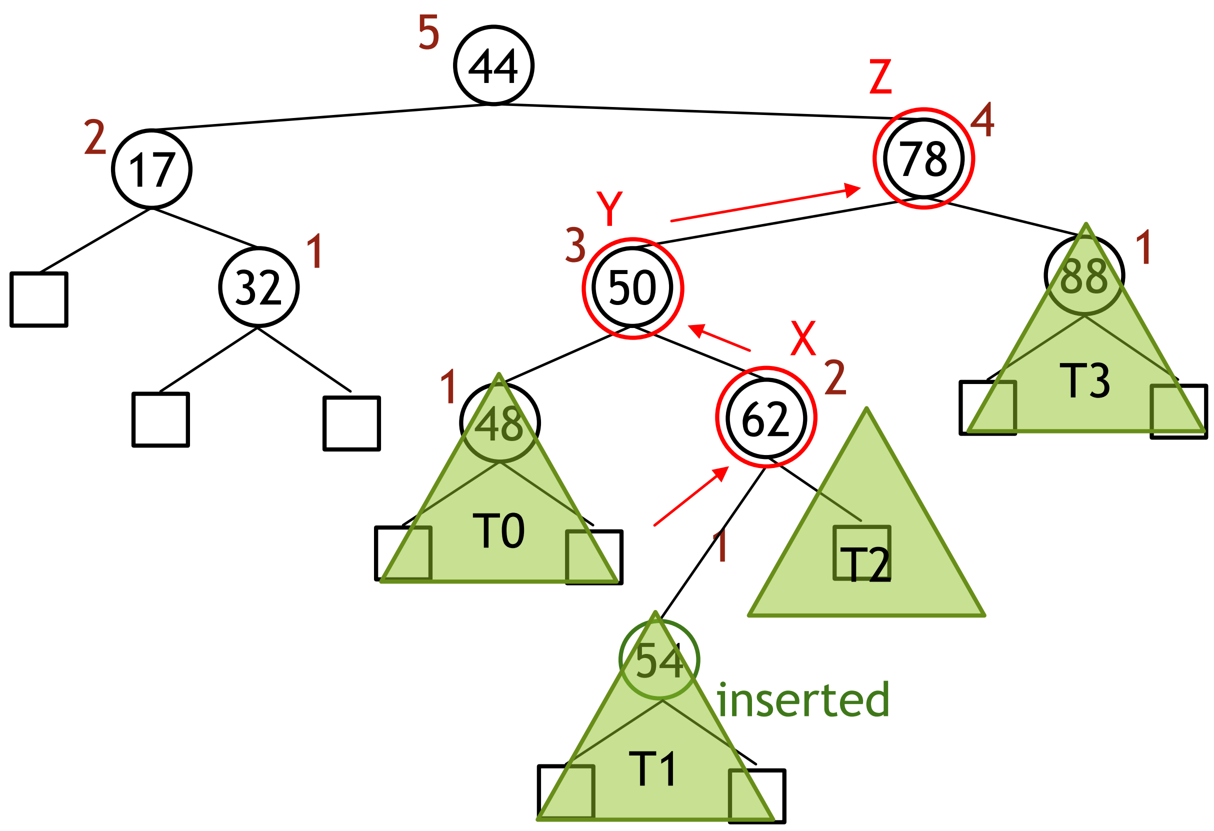


Review AVL Tree

2) Rebalancing

step 3: reorder the sequence X, Y, Z by their **inorder traversal**.

X, Y, Z -> Y, X, Z



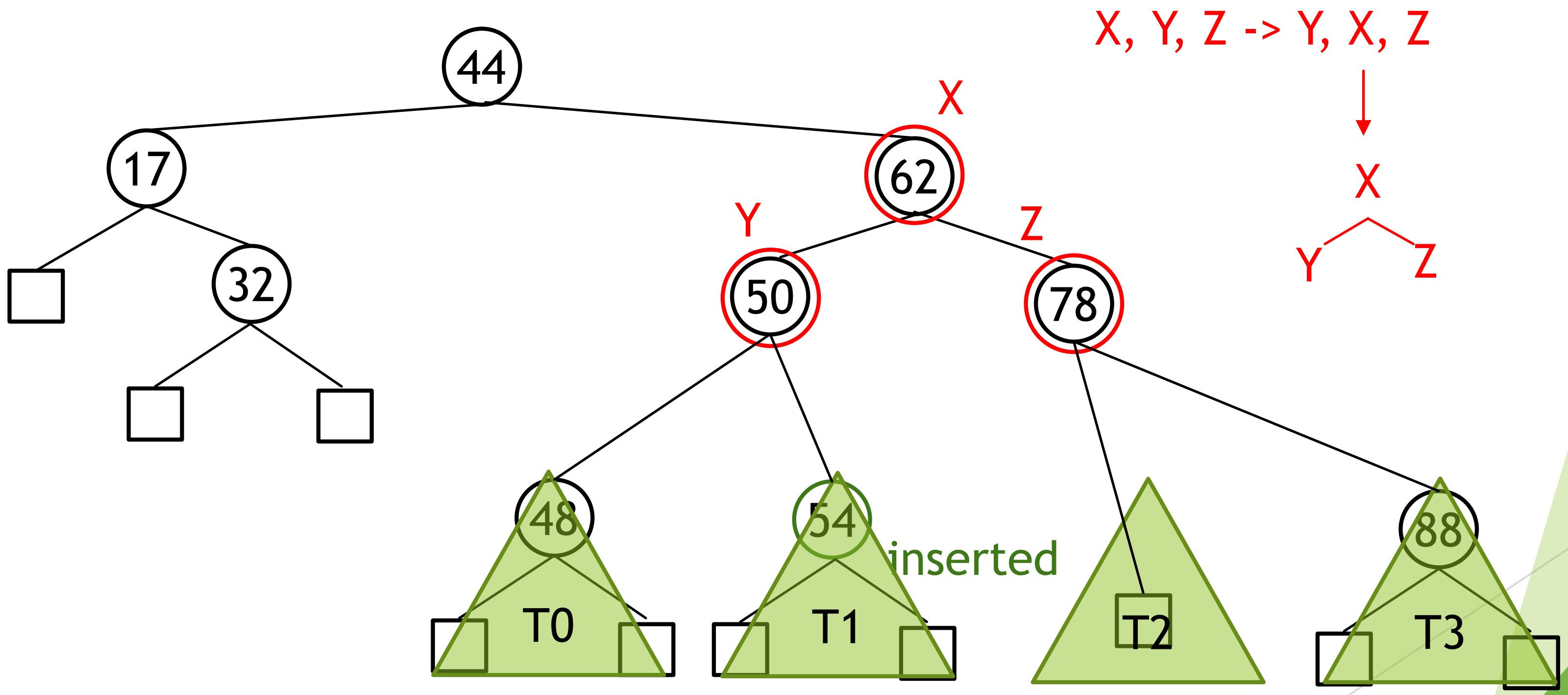
Review AVL Tree

2) Rebalancing

step 4: according to the new order, rebuilt the subtree.

three node are assigned as: left child, node, right child

connect the subtree T0, T1, T2, T3 to the new subtree (keep the same order)



Review AVL Tree

Remove

1) Remove a node according to rule of binary search tree

2) Rebalancing

   label nodes:

      Z: as the first unbalanced node

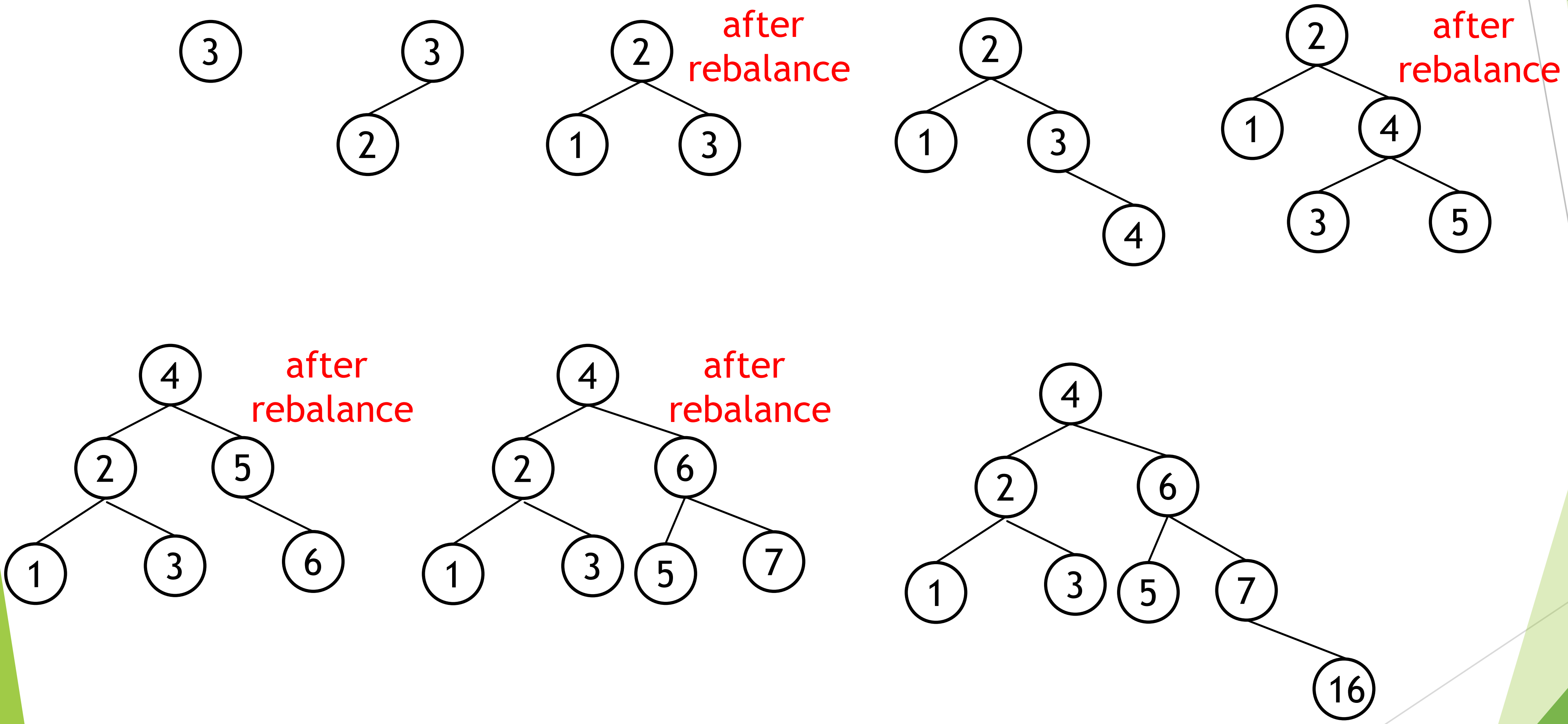
      Y: the child of Z with the larger height

      X: the child of y with the larger height

   follow the same rule of rebalancing in ‘insert’ operation

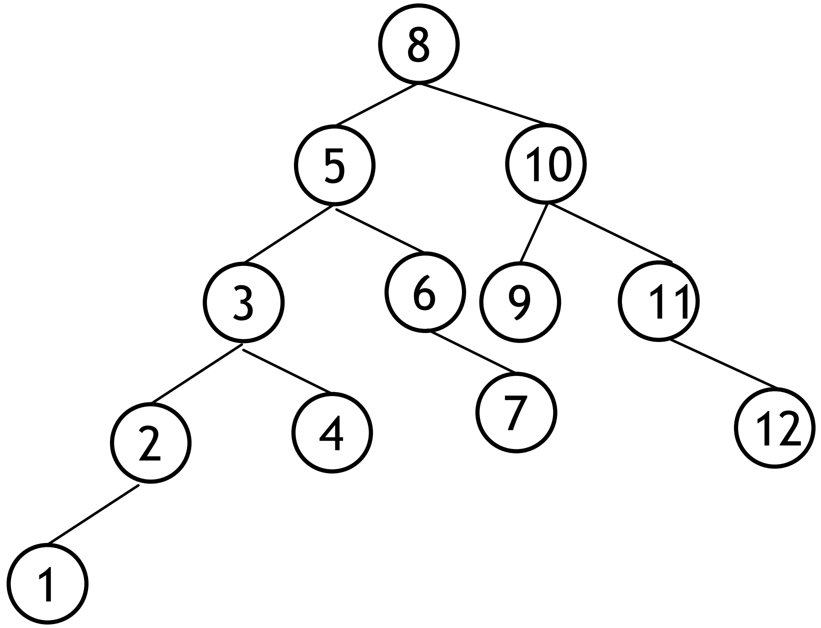
Exercise:

1. Insert the following keys one by one into an empty AVL tree: 3, 2, 1, 4, 5, 6, 7, 16
show all the steps and apply rebalance when needed.



Exercise

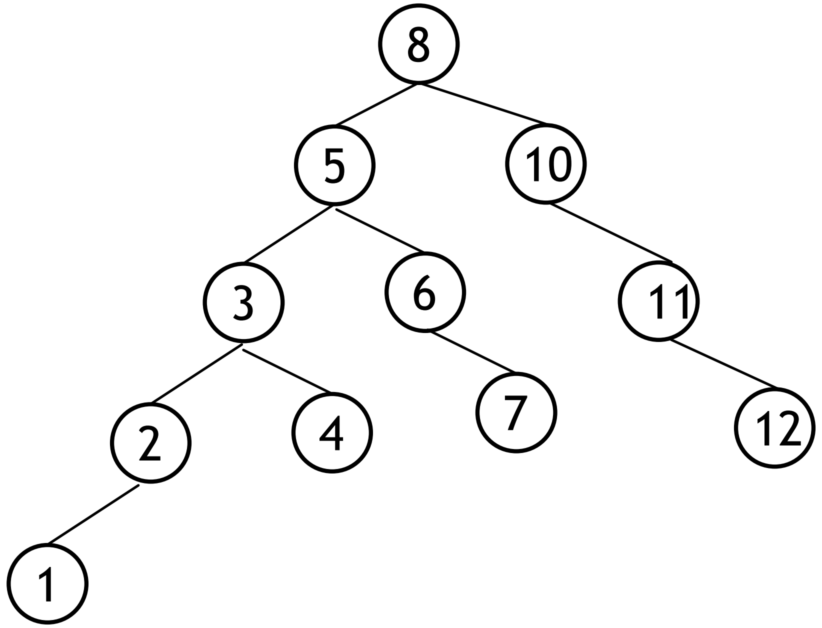
2. Consider the AVL tree below, show the tree after delete 9.



Exercise

2. Consider the AVL tree below, show the tree after delete 9.

1) delete the node

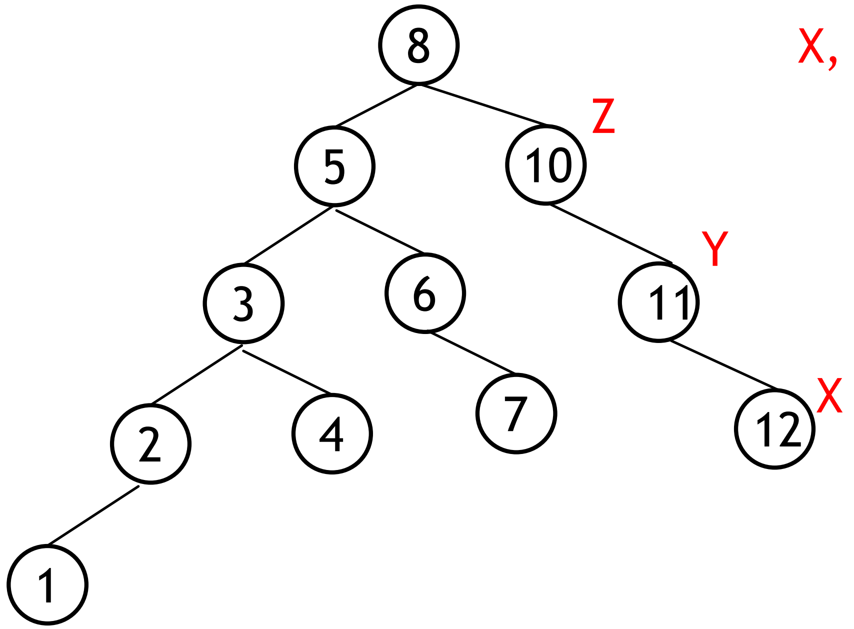


Exercise

2. Consider the AVL tree below, show the tree after delete 9.

2) rebalance

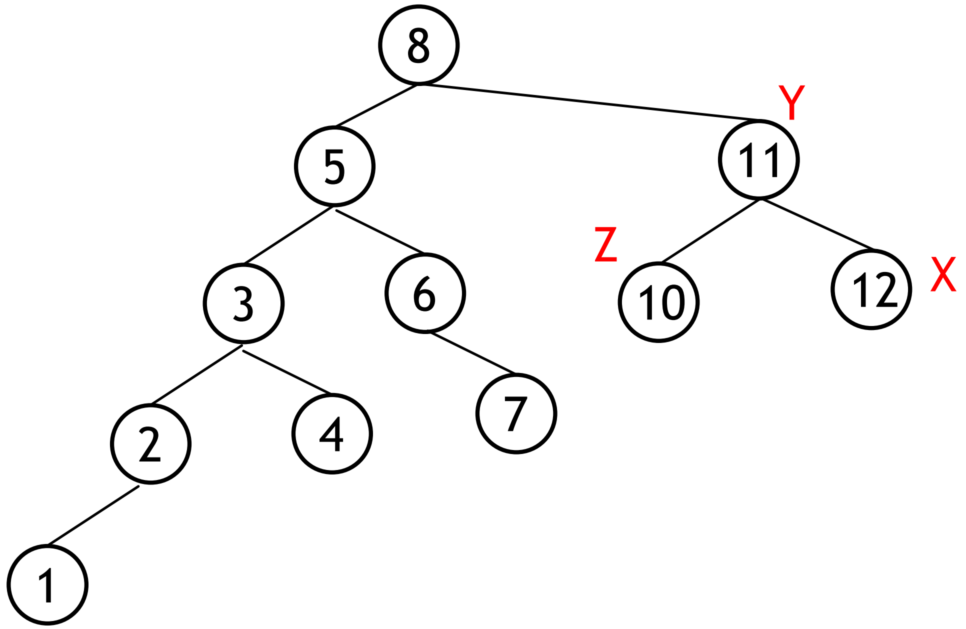
X, Y, Z -> Z, Y, X



Exercise

2. Consider the AVL tree below, show the tree after delete 9.

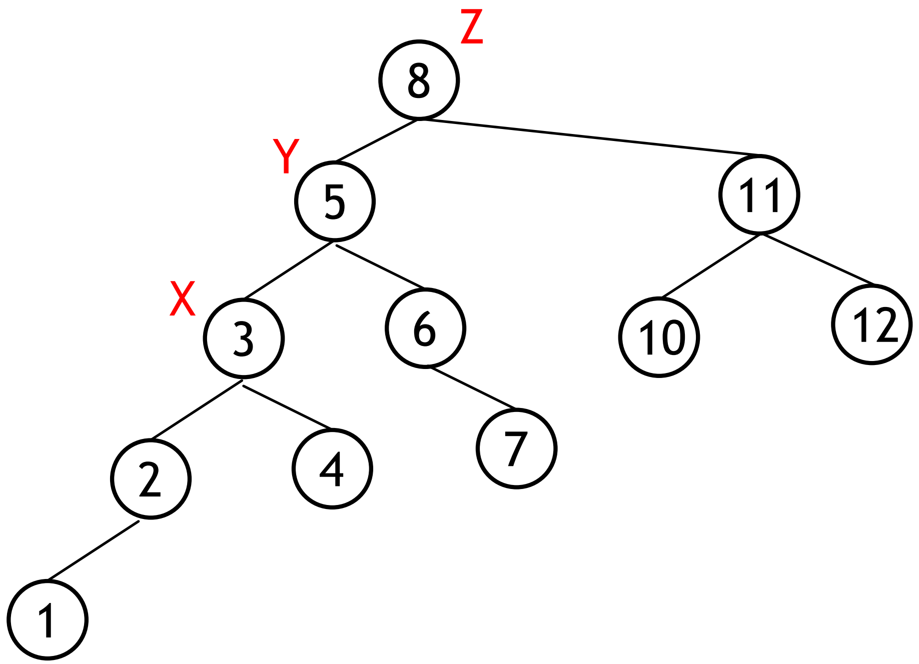
2) rebalance



Exercise

2. Consider the AVL tree below, show the tree after delete 9.

3) Rebalance again

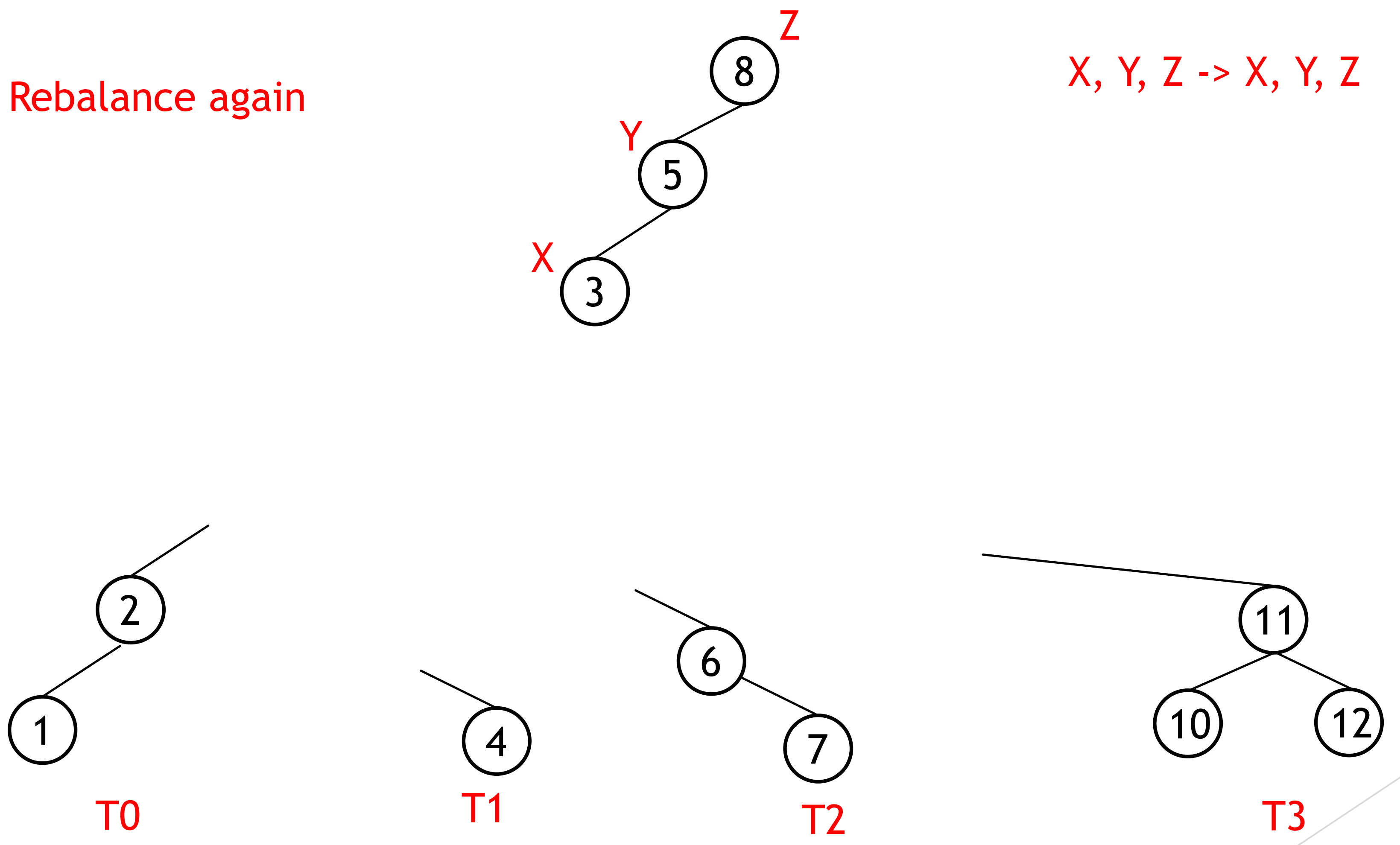


Exercise

2. Consider the AVL tree below, show the tree after delete 9.

3) Rebalance again

X, Y, Z -> X, Y, Z

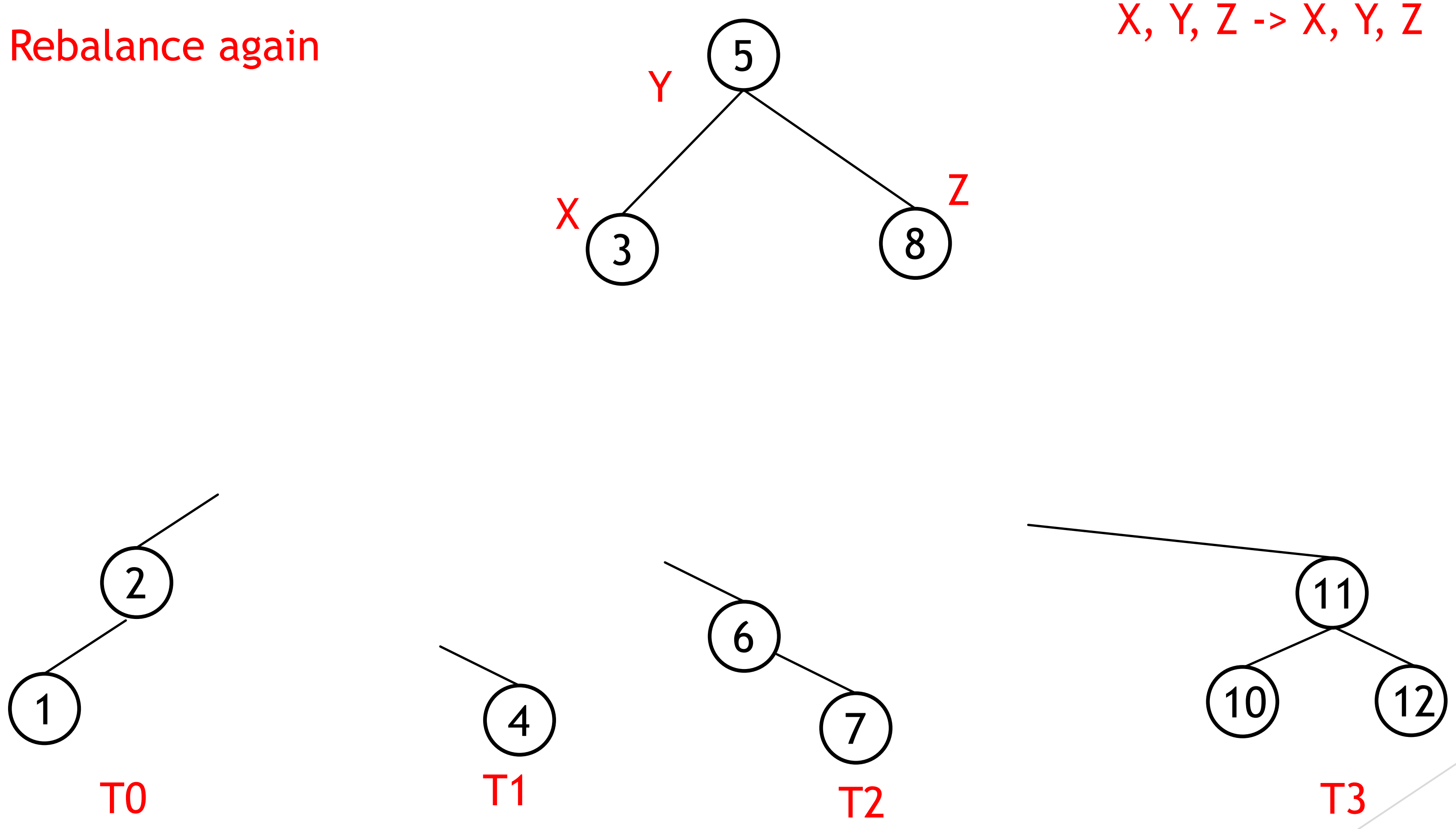


Exercise

2. Consider the AVL tree below, show the tree after delete 9.

3) Rebalance again

X, Y, Z -> X, Y, Z

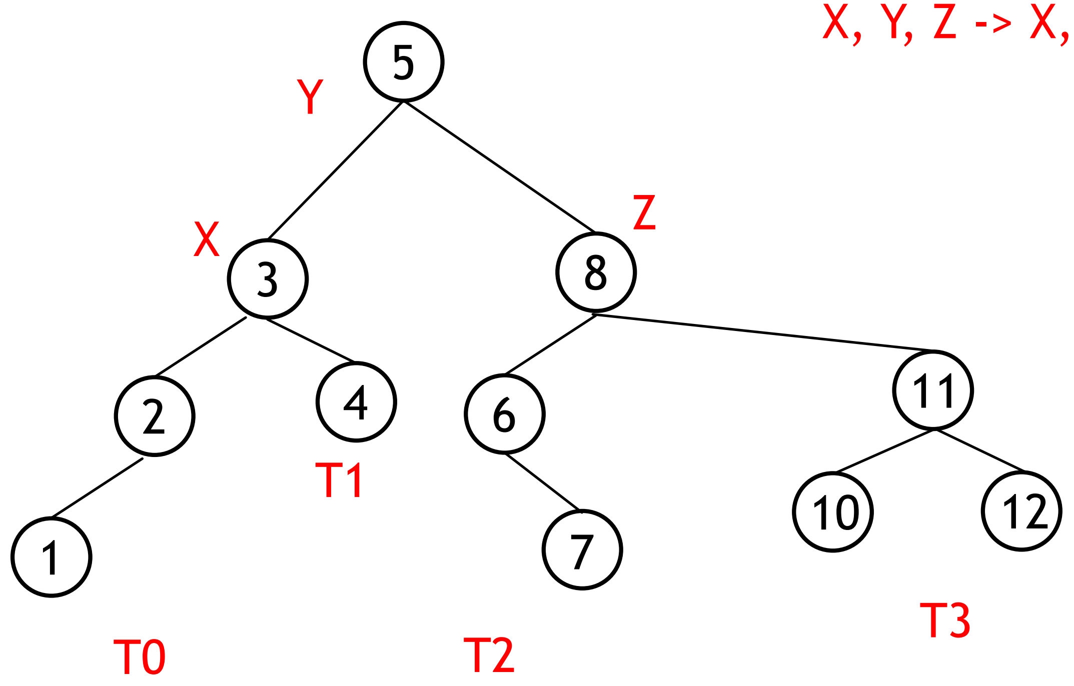


Exercise

2. Consider the AVL tree below, show the tree after delete 9.

3) Rebalance again

X, Y, Z -> X, Y, Z

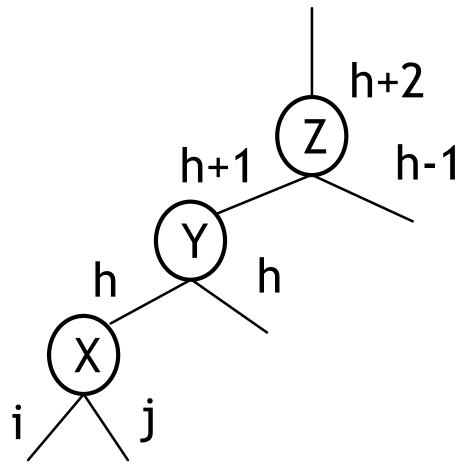


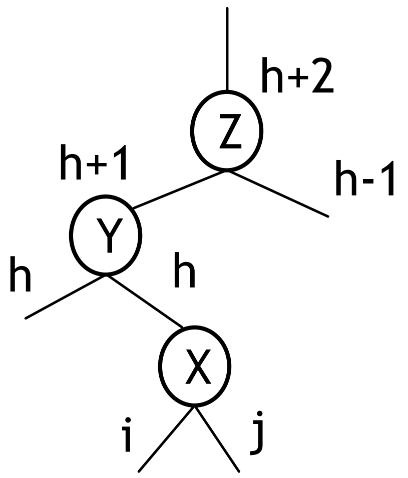
Exercise (R11.11)

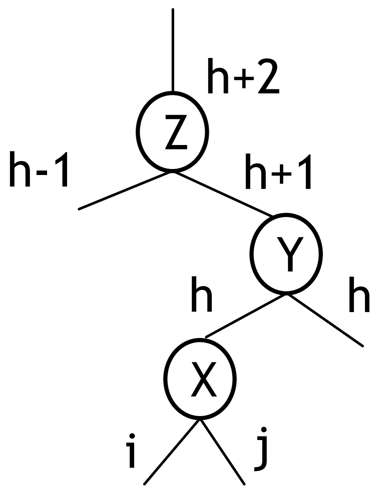
3. Consider a deletion operation in an AVL tree that triggers a trinode restructuring
For the case in which both children of the node denoted as y have equal heights.
Give a schematic figure showing the tree before and after deletion. What is the net
Effect of the height of the rebalanced subtree due to the operations?

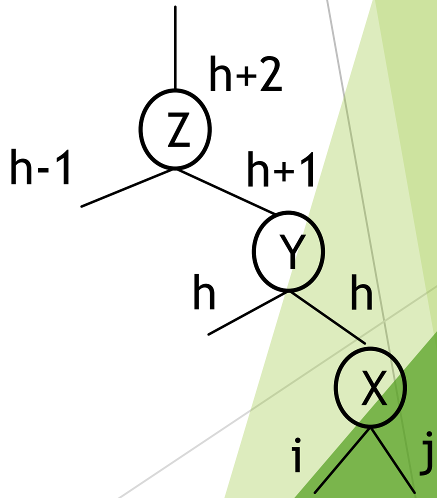
(After Deleting)
Before
Rebalance

Subtree start by X, Y, Z should be
higher or equal to their
neighbor subtree







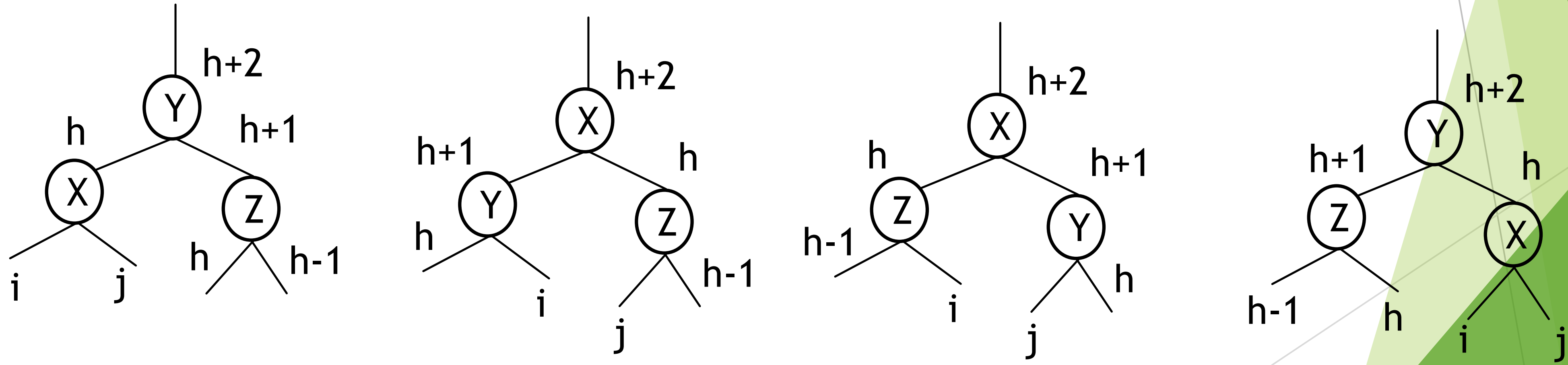


Exercise (R11.11)

3. Consider a deletion operation in an AVL tree that triggers a trinode restructuring For the case in which both children of the node denoted as y have equal heights. Give a schematic figure showing the tree before and after deletion. What is the net Effect of the height of the rebalanced subtree due to the operations?

After Rebalance

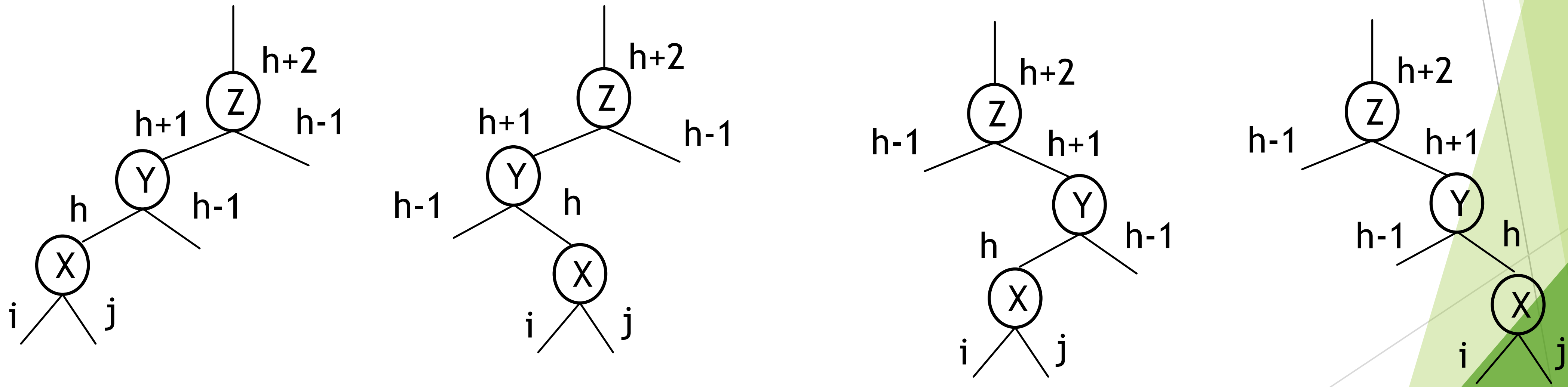
No change to the height



Exercise (R11.12)

4. Repeat the previous problem, considering the case in which y’s children Start with different heights (before deleting).

(After Deleting)
Before
Rebalance

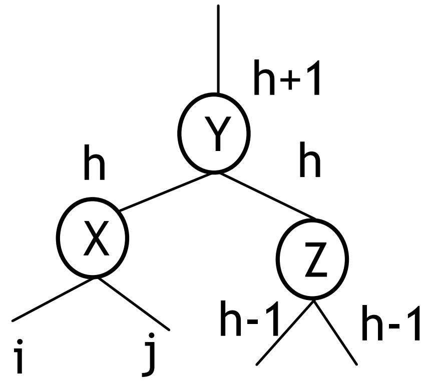


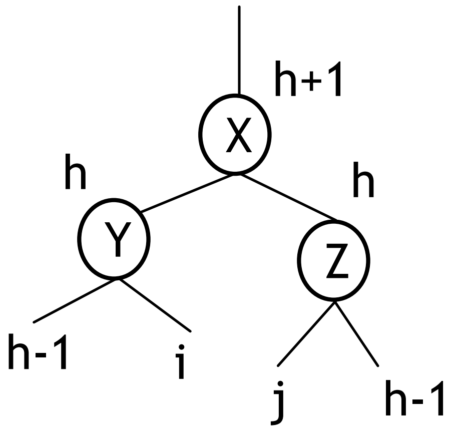
Exercise (R11.12)

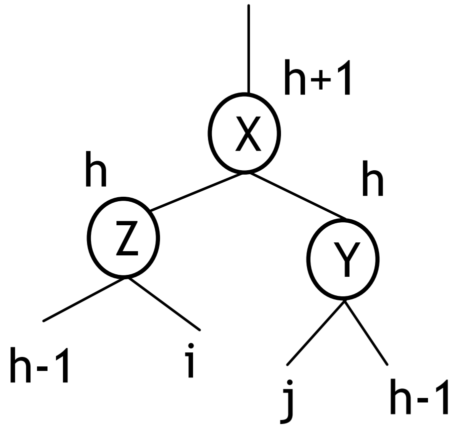
4. Repeat the previous problem, considering the case in which y’s children Start with different heights (before deleting).

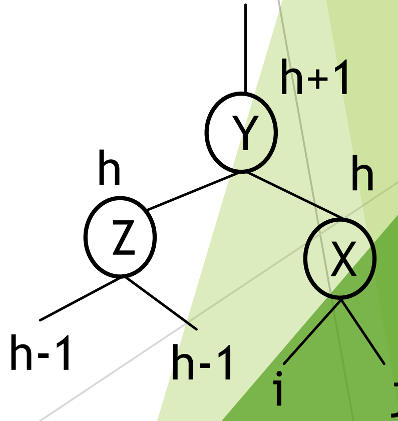
After Rebalance

Height - 1









Exercise C-11.39

Raw a schematic of an AVL tree such that a single remove operation could
Require Ω(logn) trinode restructurings (or rotations) from a leaf to the root
In order to restore the height-balance property.