

CS2700: PROGRAMMING AND DATA STRUCTURES.

BINARY SEARCH TREES

(HEIGHT BALANCED TREES)

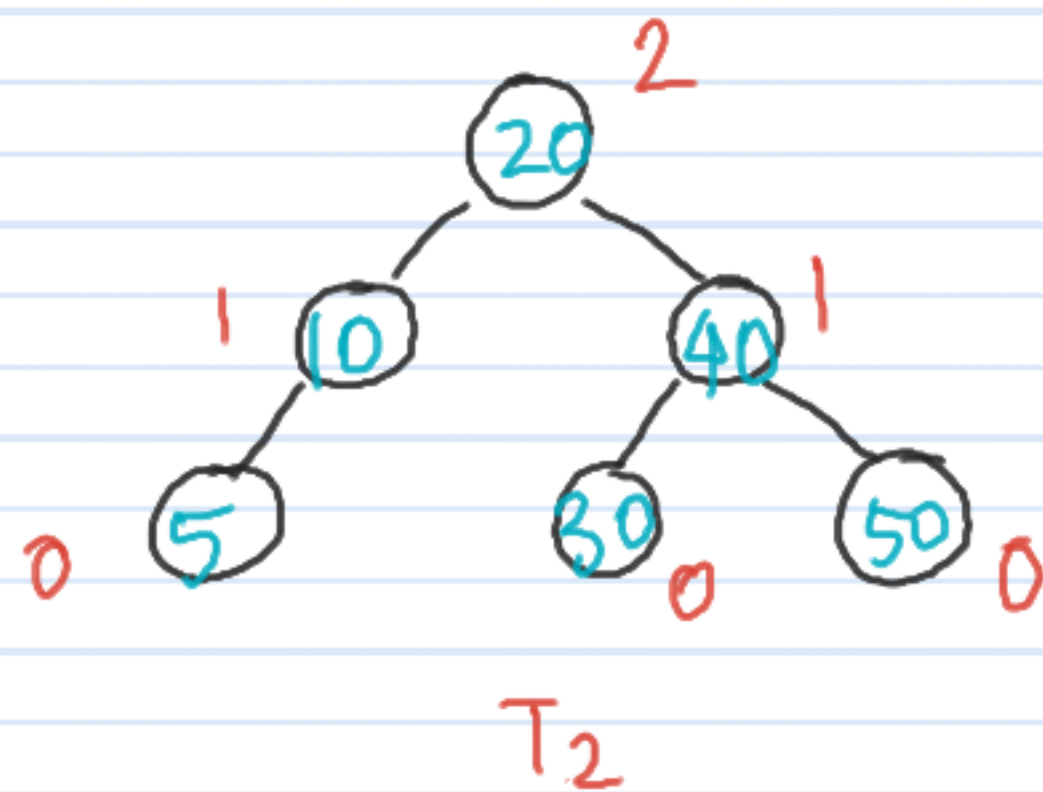
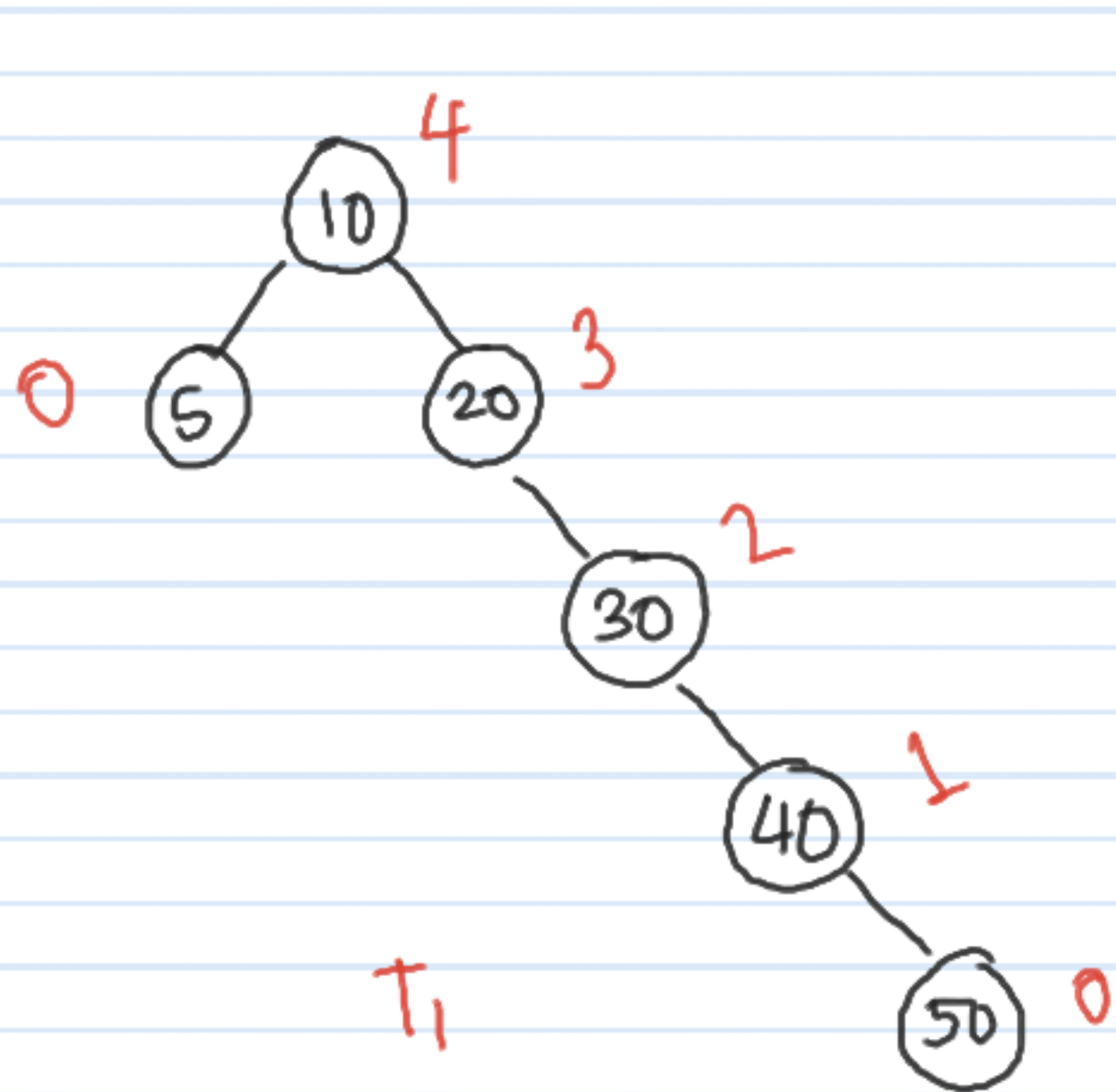
GOALS:

- WHAT IS HEIGHT BALANCE?
- HOW TO ACHIEVE IT?

BINARY SEARCH TREES

- COMPLEXITY OF OPERATIONS :
 - depends on height of tree
- Height of BST depends on order of insertions and deletions
- All operations in a BST take _____ time (insert, delete, find Min, search) in worst case

DEEP VERSUS SHALLOW TREES

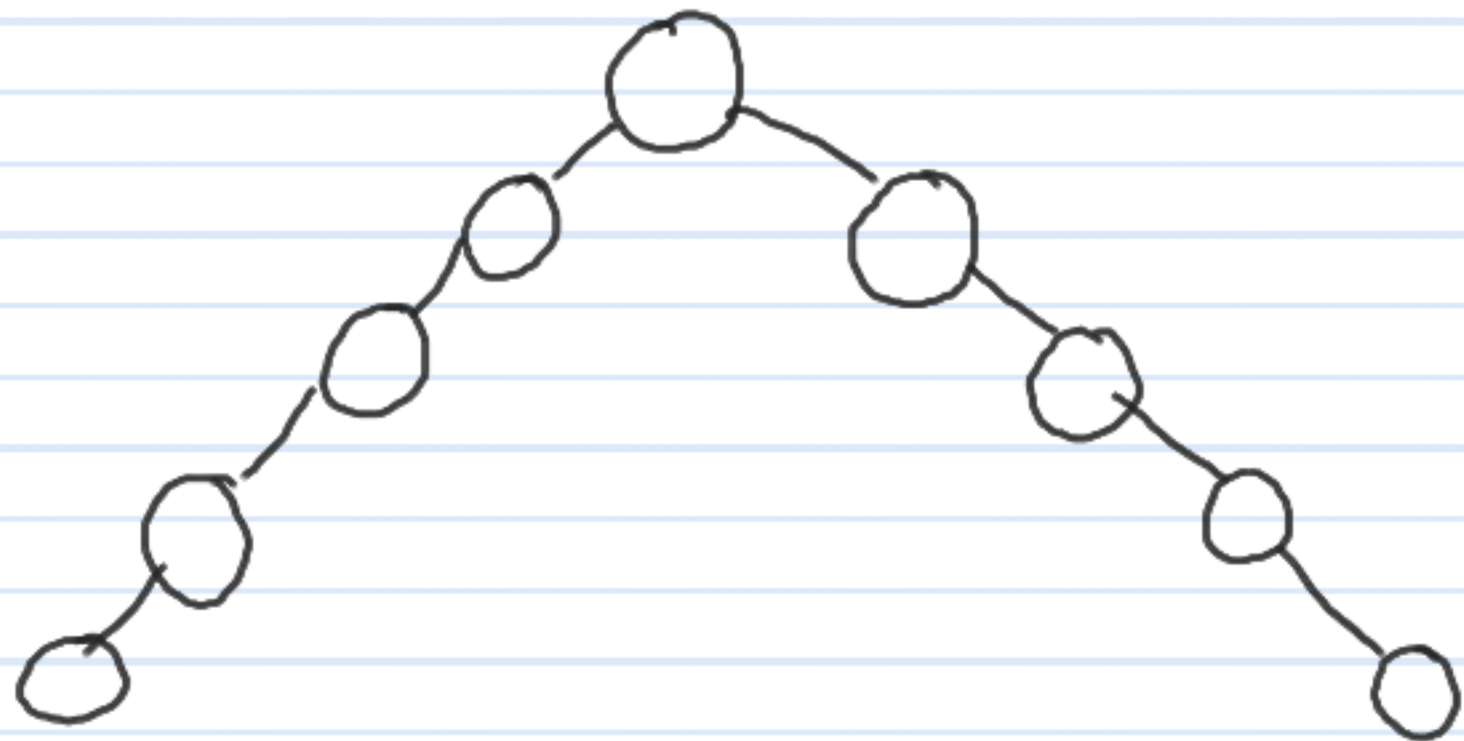


- can we fill T_2 with keys in T_1 ?
- is there any specific property of T_1 that enables it?

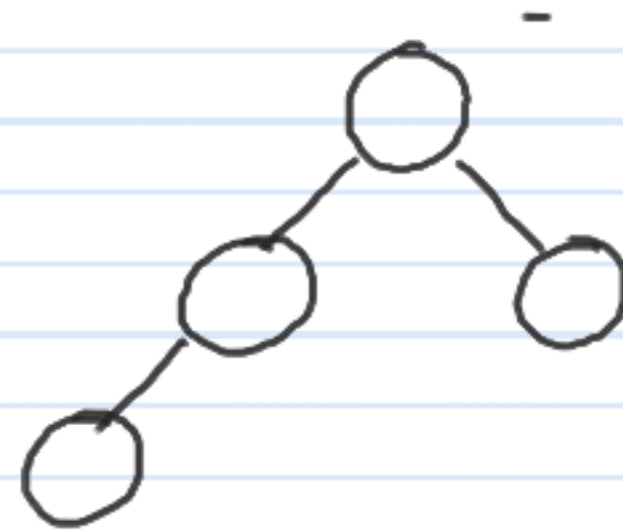
HEIGHT BALANCED TREES

— HOW DO WE DEFINE BALANCED TREES?

(1) ENSURE THAT HEIGHT
OF LEFT and RIGHT SUBTREES
ARE SAME



(2) ENSURE THIS PROPERTY
AT EVERY NODE IN THE TREE



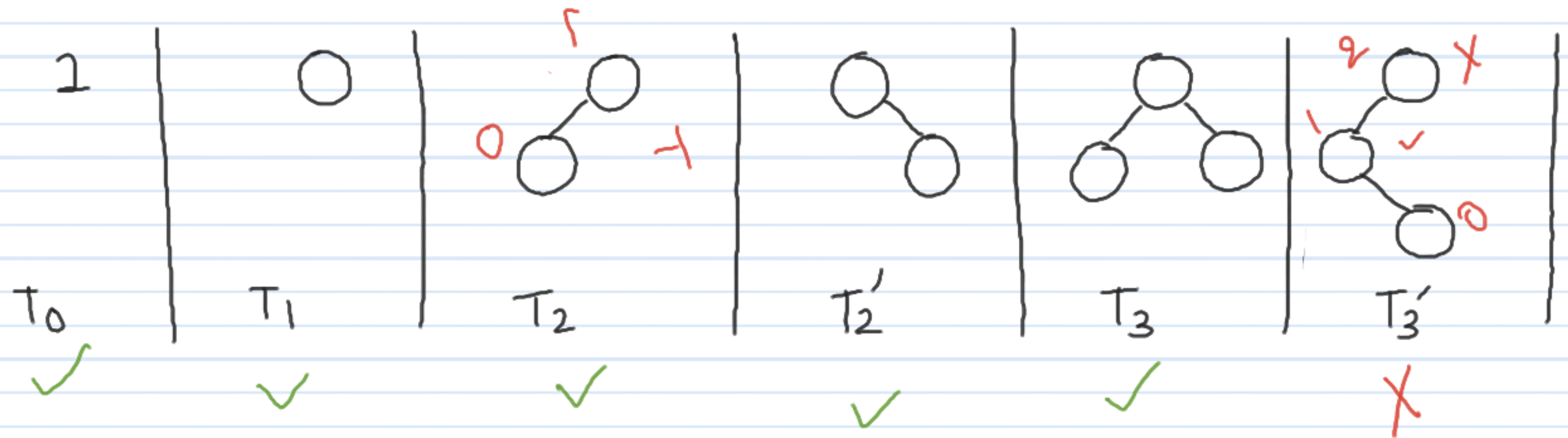
A 4 node
tree cannot
exist!!

(too rigid a
property)

HEIGHT BALANCED TREES

[AVL TREES]

- AT every node height difference can be at most 1.

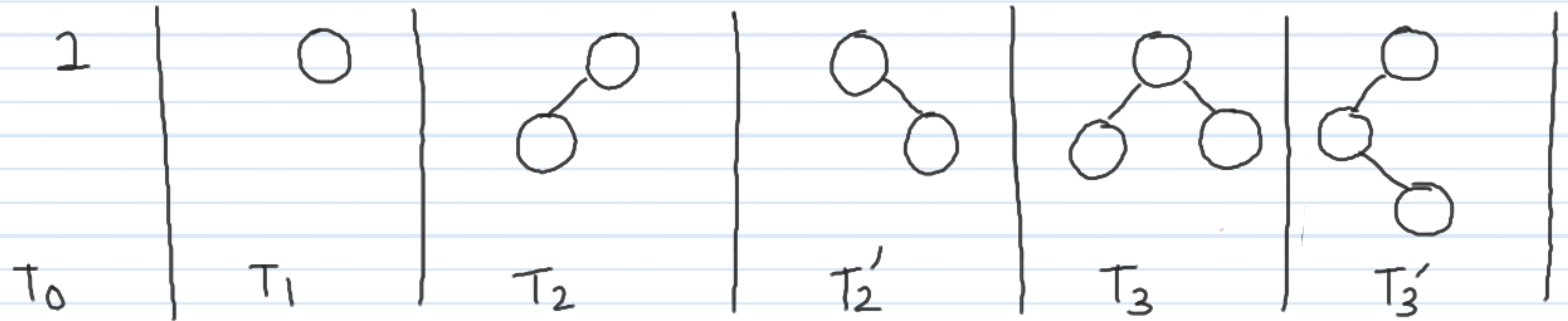


HEIGHT BALANCED TREES

[AVL TREES]

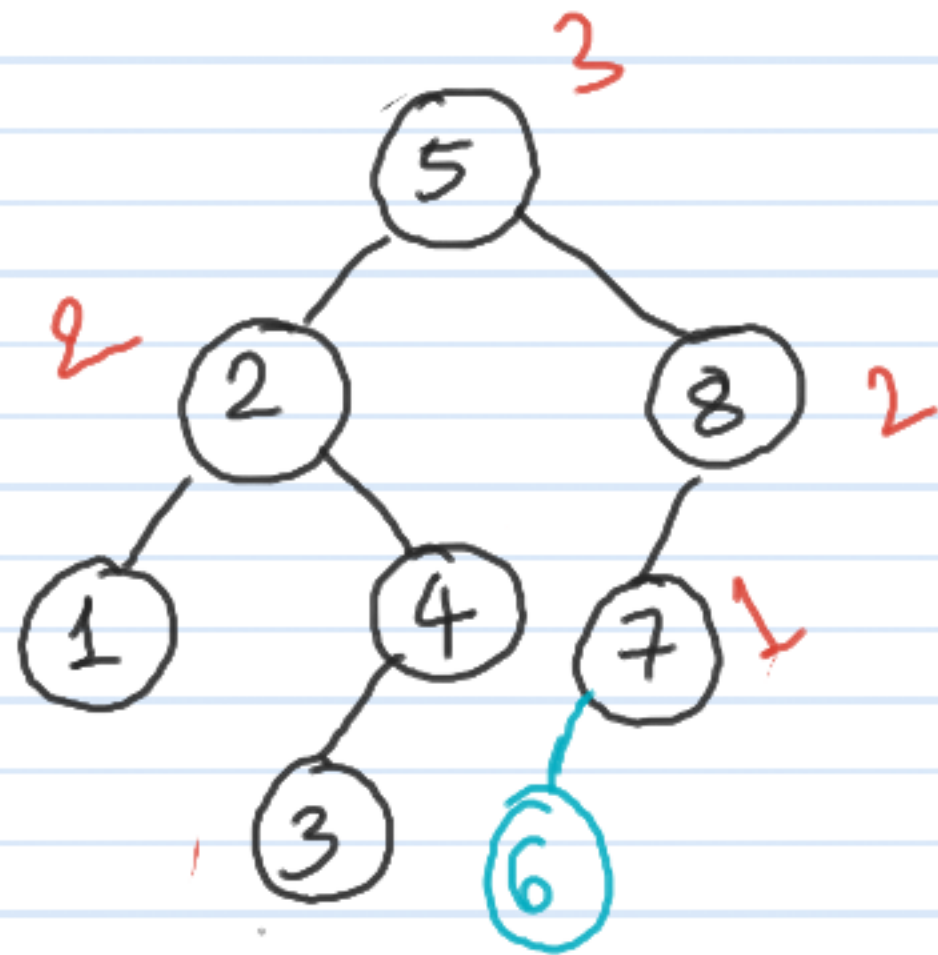
- AT every node height difference can be at most 1.

Adelson Velski Landis



HOW TO MAINTAIN HEIGHT BALANCE?

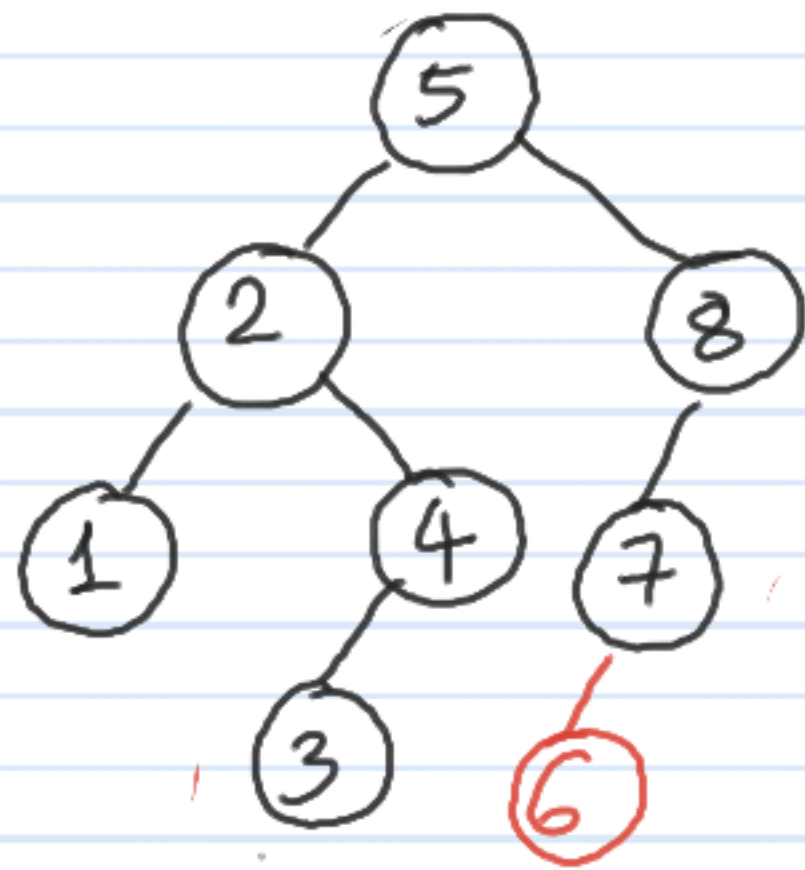
Insertion can violate height balance



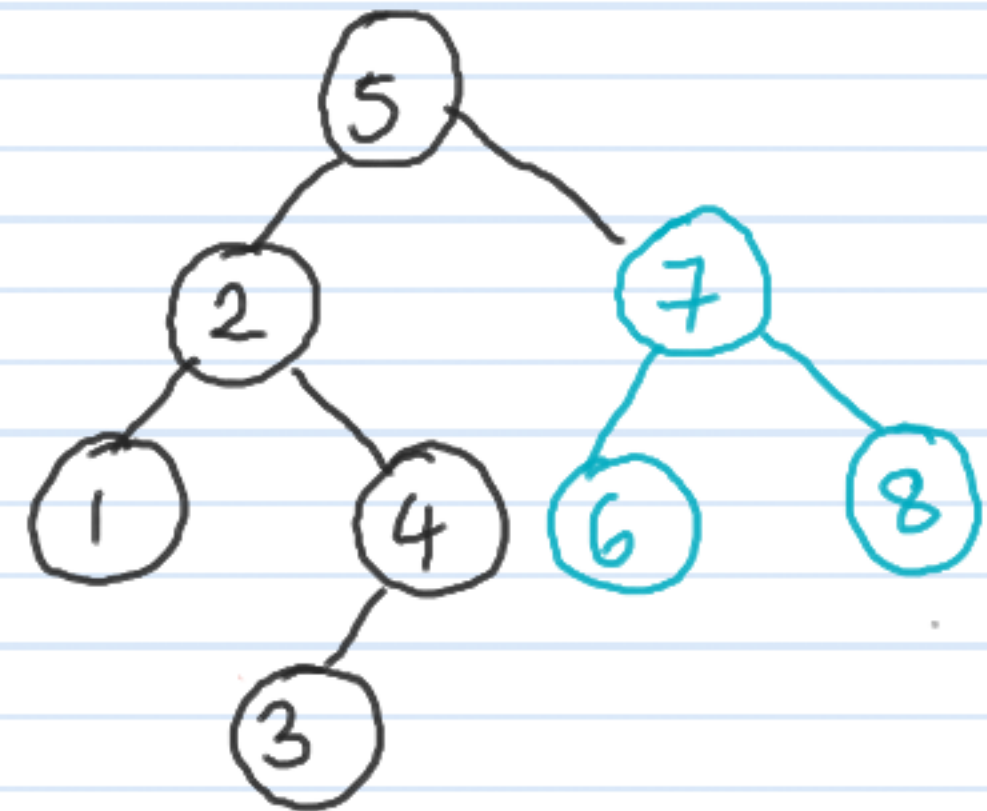
insert 6

HOW TO MAINTAIN HEIGHT BALANCE?

Insertion can violate height balance



uses rotation to restore height balance

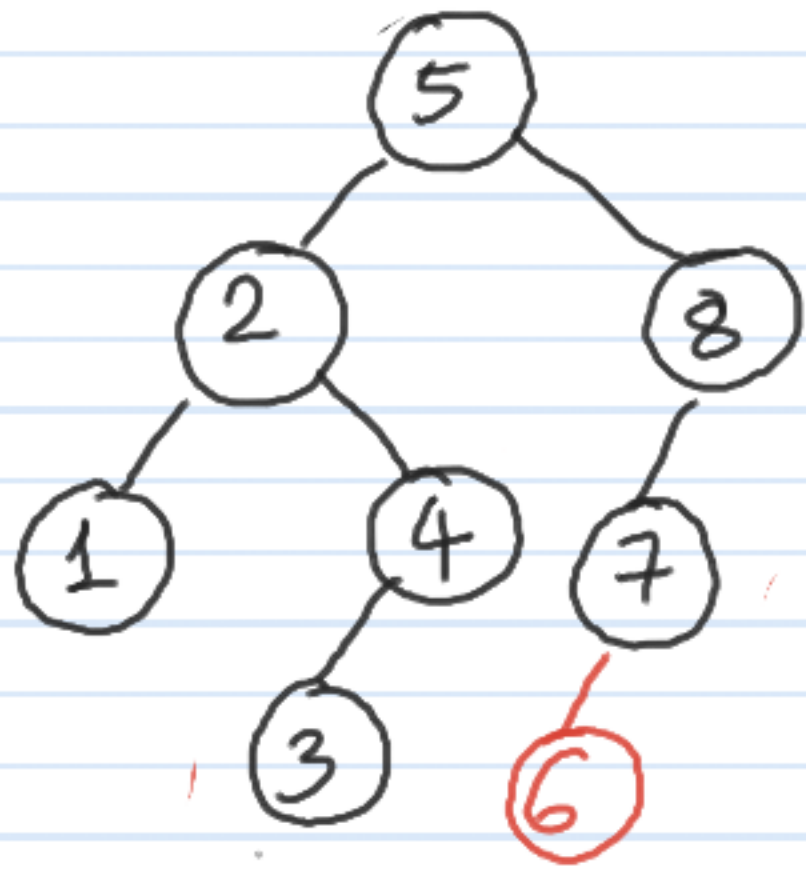


HOW TO MAINTAIN HEIGHT BALANCE?

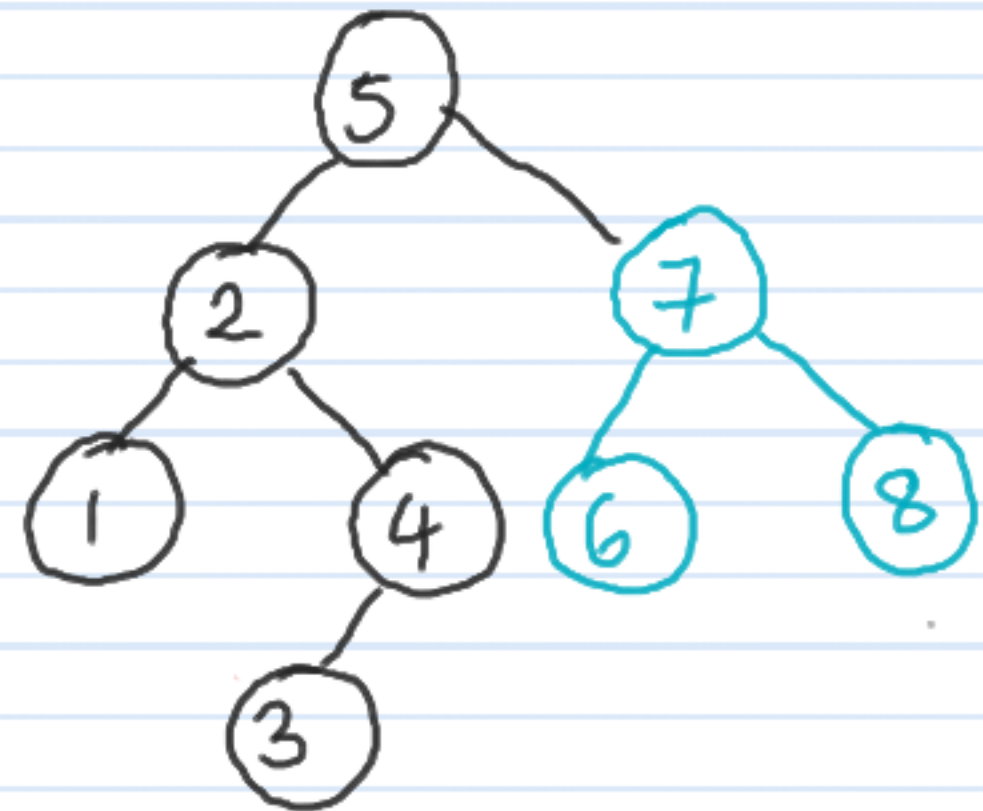
Insertion can violate height balance

- multiple BST shapes can fit some keys

- BST property must be respected

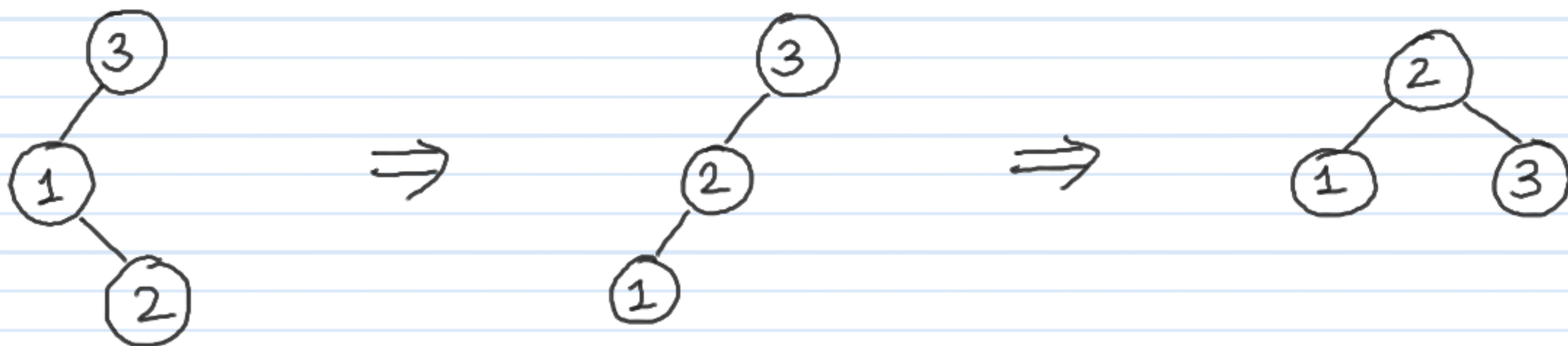


uses rotation to restore height balance



- Only nodes on path from root to inserted node are affected.

ROTATION another example



- What is a rotation?
- How to identify which type / how many rotations are needed?
- How many are sufficient in worst case?

INSERTION INTO AN AVL TREE

Recall

- Height balance definition
- Search property has to be respected.
- α be the **deepest node** which is imbalanced.

4 cases:

(1) Left left

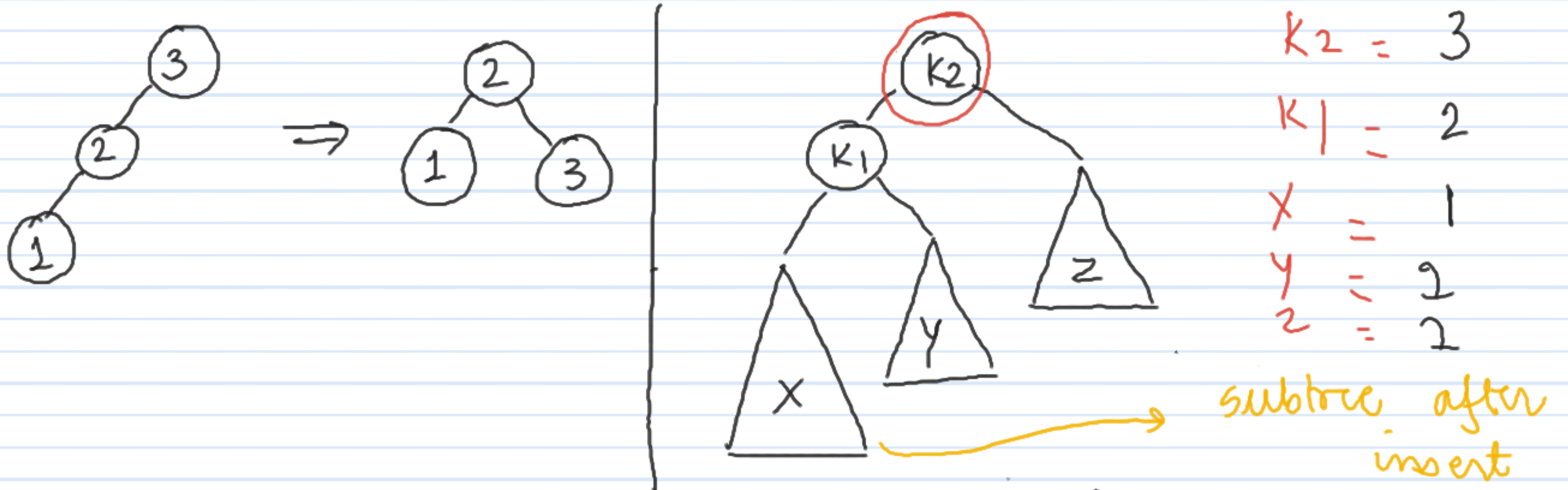
(2) Left Right

(3) Right-left

(4) Right Right

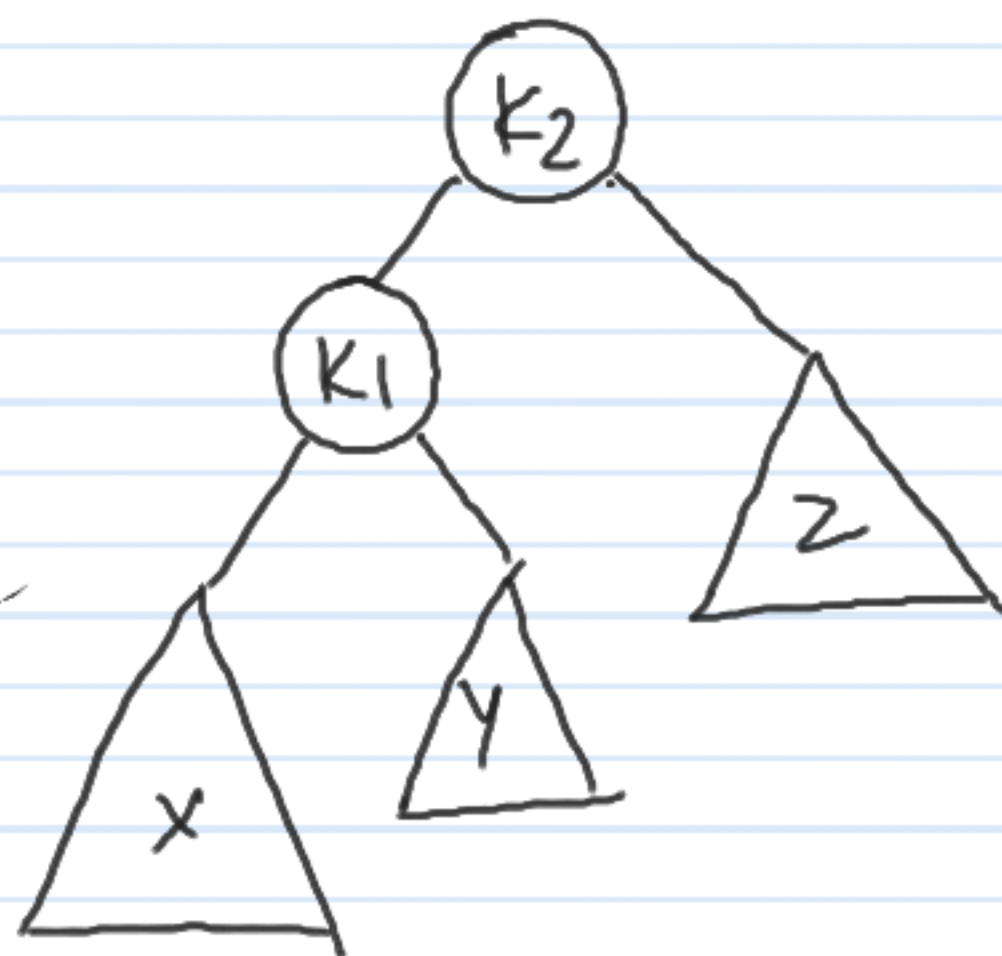


AVL TREE INSERTION : case 1 (left-left)

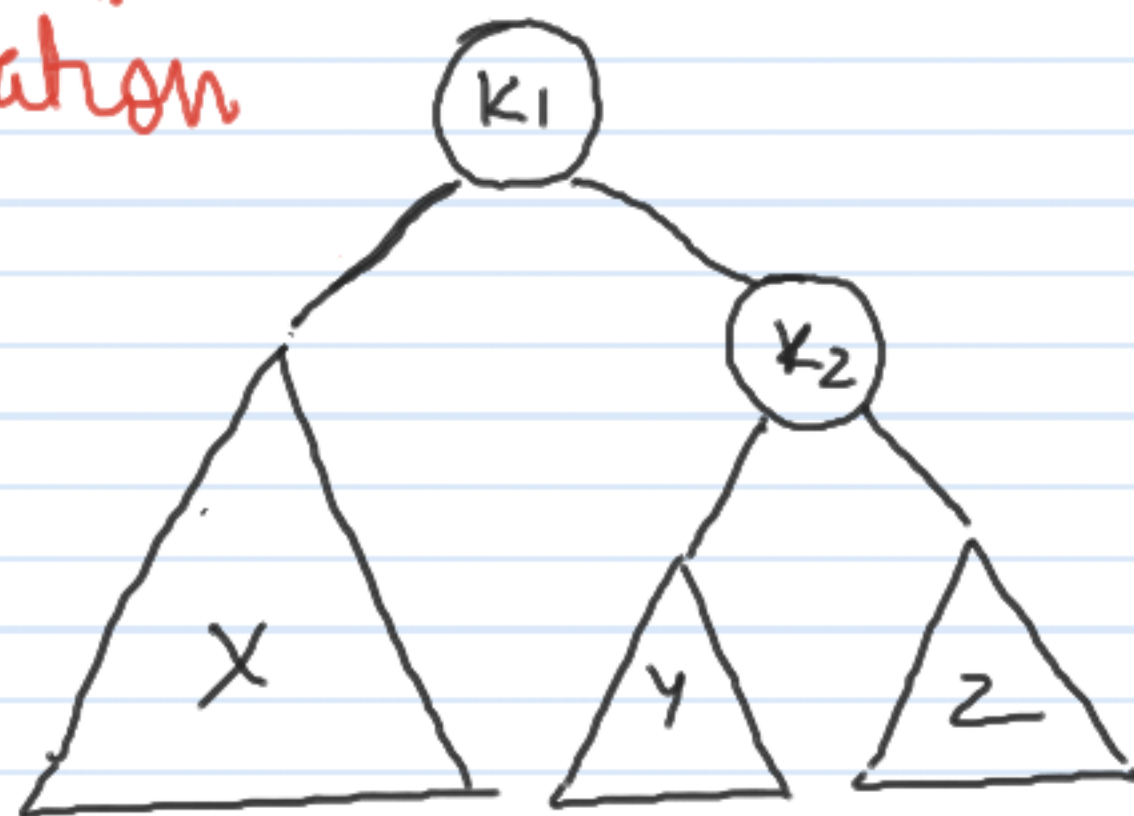


- K_2 is the deepest node with imbalance
- $h(K_2 \rightarrow \text{left}) - h(K_2 \rightarrow \text{right}) > 1$
- K_1 is height balanced.
- Can X (after insert) and Y be of same height?

AVL TREE INSERTION : case 1 (left-left)



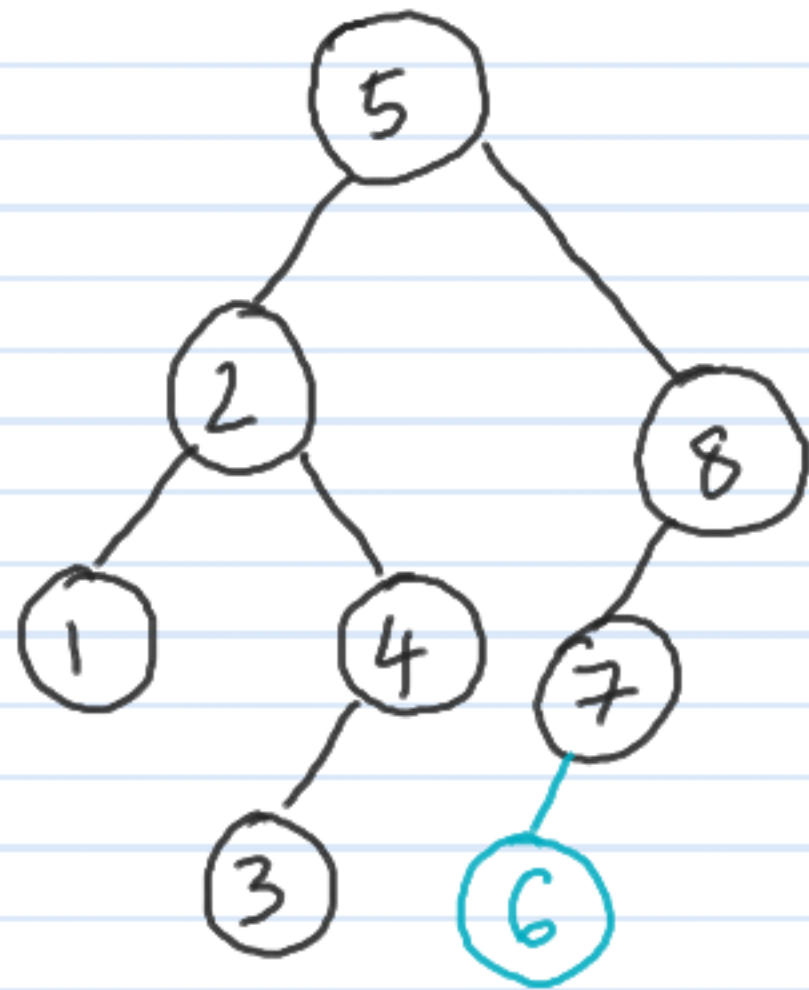
right rotation



BST order?

- Pull K_1 and rotate it (shake it) and let gravity (bst property) take care.

• $X \uparrow$ $Y \text{ — }$ $Z \downarrow$



insert 6

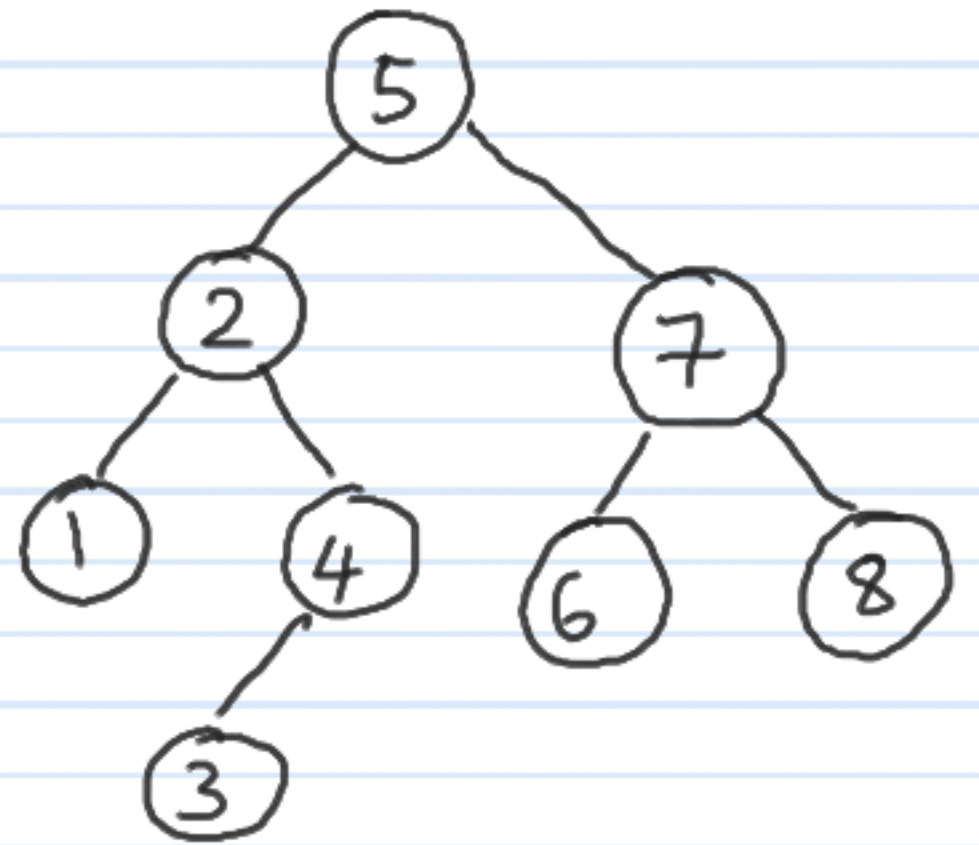
$$k_1 = 7$$

$$k_2 = 8$$

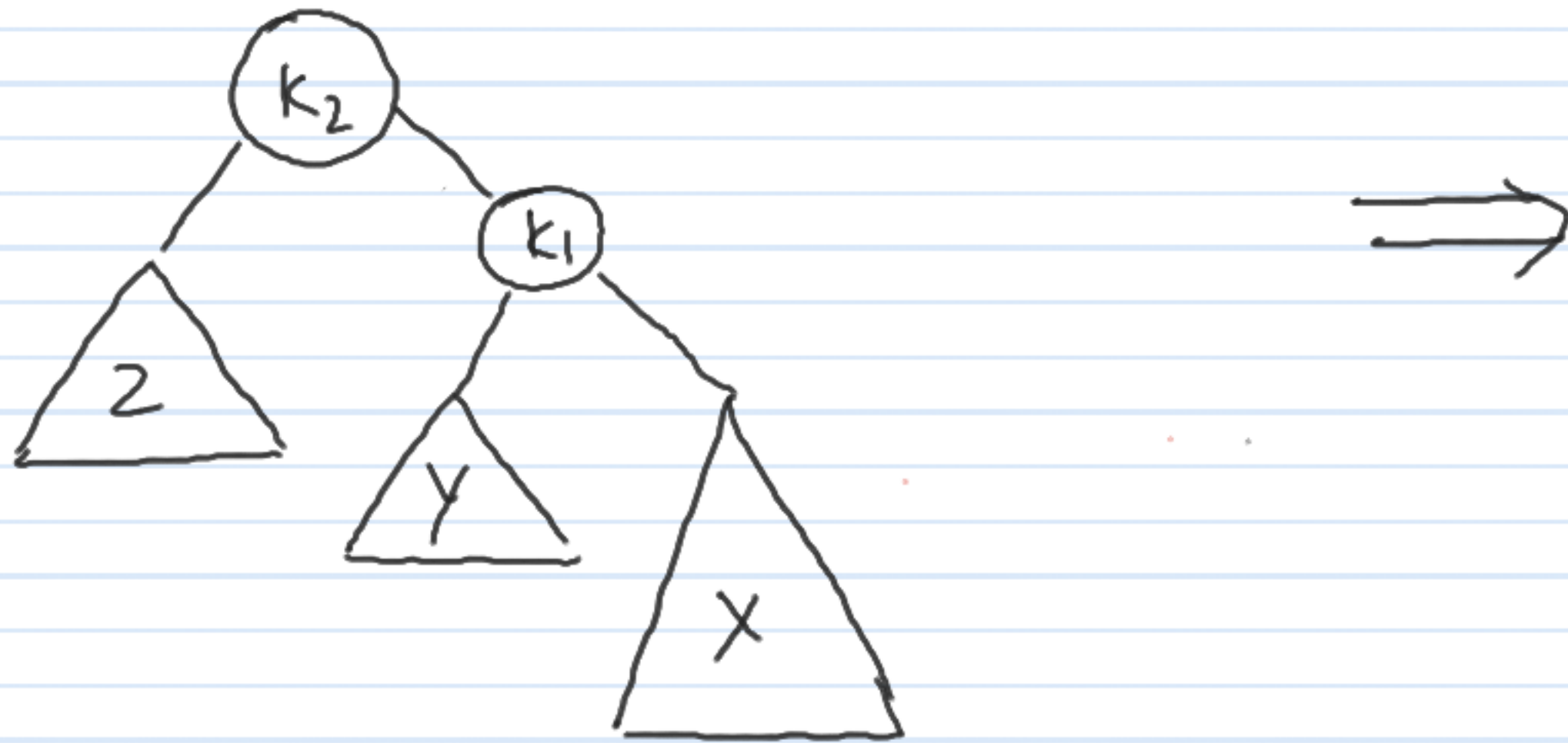
$$X = 6$$

$$Y = 2$$

$$Z = 7$$



AVL TREE INSERT : CASE 4 (Right-Right)



subtree x
after insert.

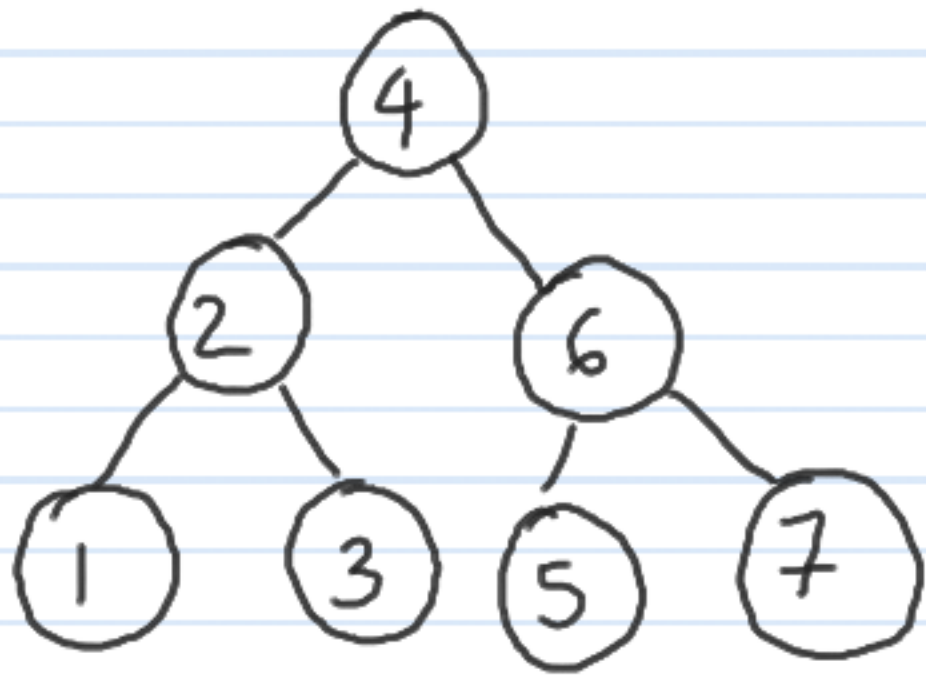
AVL TREE INSERT : EXAMPLE,

insert 3, 2, 1, 4, 5, 7

AVL TREE INSERT : EXAMPLE.

insert 3, 2, 1, 4, 5, 7

16, 15, 14, 13, 12, 11, 10, 9, 8



AVL TREE INSERT : EXAMPLE, (Case 3 Right left)

insert 3, 2, 1, 4, 5, 7

16, 15, 14, 13, 12, 11, 10, 9, 8

