

Trees Homework 1: BST, AVL

March 29, 2019

1 Binary Search Tree

Draw the binary search tree obtained when the keys 1, 2, 3, 4, 5, 6, 7 are inserted in the given order into an initially empty tree. What is the problem of the tree you get? Why is it a problem? How could you modify the insertion algorithm to solve this problem. Justify your answer. [10 pts]

2 AVL Tree

2.1

- i. Insert the following sequence of elements into an AVL tree, starting with an empty tree: 10, 20, 15, 25, 30, 16, 18, 19. [15 pts]
- ii. Delete 30 in the AVL tree that you got. [5 pts]

2.2

Draw an AVL tree that satisfies the following three conditions [10 pts]:

- i. The tree has exactly 11 nodes.
- ii. There are no pair of nodes that, if extracted one after the other, will cause the height decrease by 1.
- iii. There is no key whose insertion will increase the height by 1.

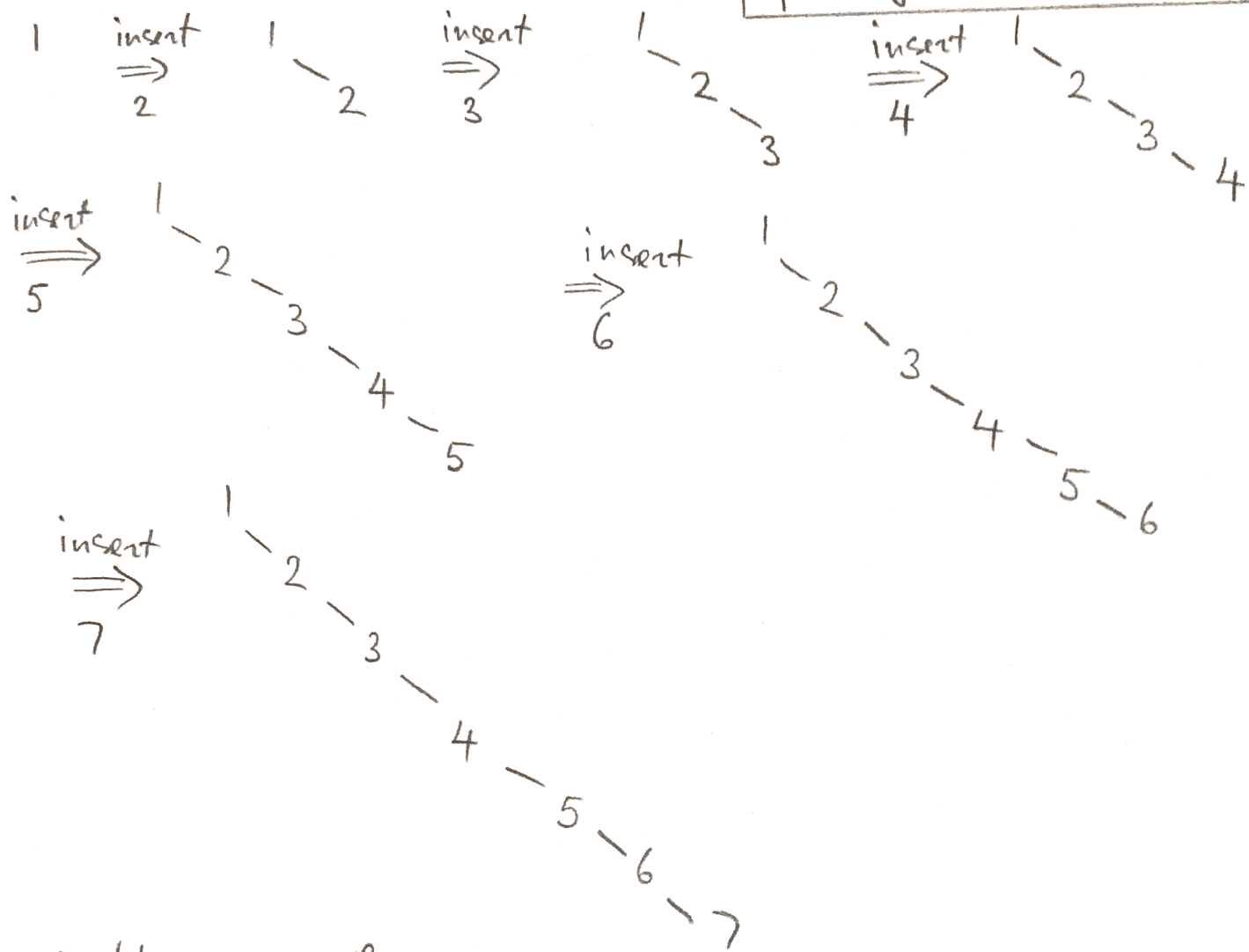
3 Tree traversals

Given the following preorder and in-order traversals for an unknown binary tree, determine the exact tree that would generate these traversals and then draw that tree. Once you have generated the tree be sure to check your work [10 pts].

Preorder: 9,5,3,1,4,8,6,20,12,10,11,30,21,31

Inorder: 1,3,4,5,6,8,9,10,11,12,20,21,30,31

① Binary Search Tree:



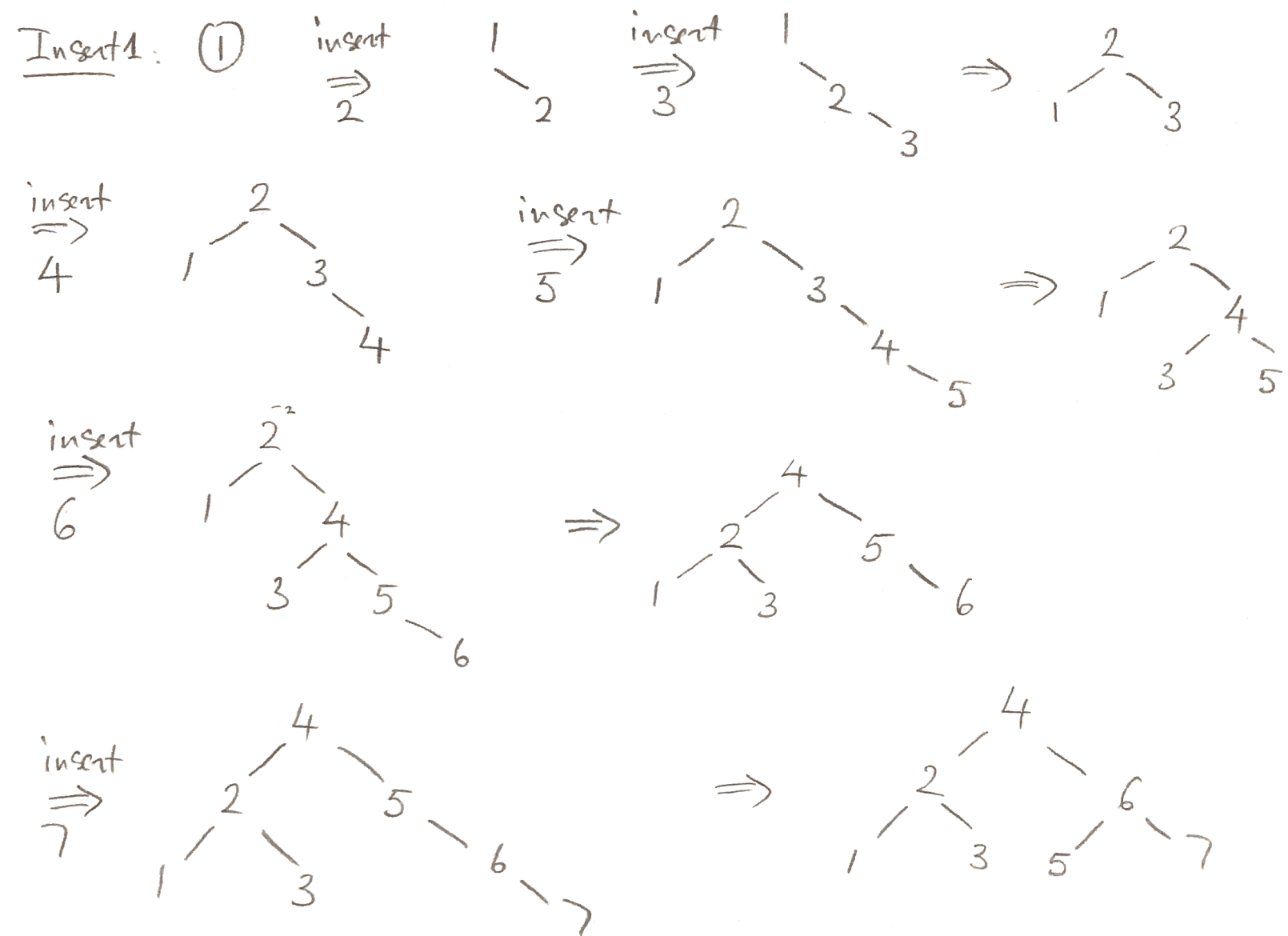
problem(s) of the tree:

- every single key is a right child of their parents
- The tree is not balanced in term of AVL principals.
- The tree is neither full nor balanced.

why:

- Balancing Factors of keys 1, 2, 3, 4, 5 violate in term of AVL.

* Modify the insertion algorithm by applying AVL theorem as it goes insertion.



② AVL

phong vo

2.1: i.

insert
⇒
10

⑩

insert
⇒
20

10
20

insert
⇒
15

10
20
15

⇒
15
10 20

insert
⇒
25

15
10 20
25

insert
⇒
30

15
10 20
25 30

⇒

15
10 25
20 30

insert
⇒
16

15
10 25
20 30
16

⇒
15
10 20
16 25 30

⇒

20
15 25
10 16 30

insert
⇒
18

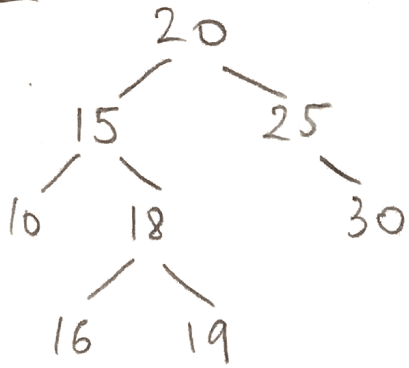
20
15 25
10 16 30
18

insert
⇒
19

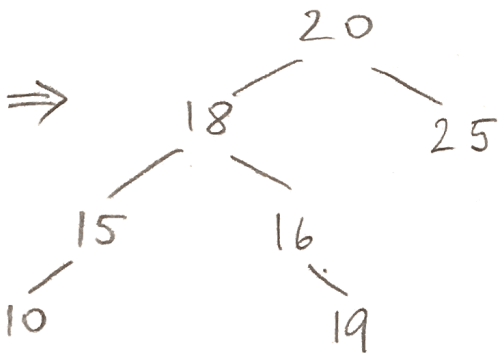
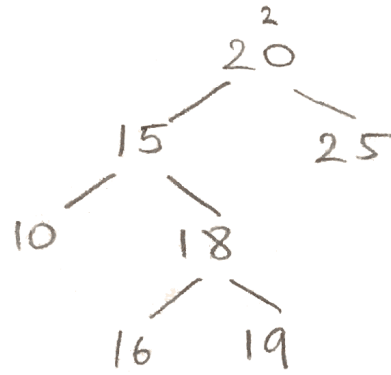
20
15 25
10 16 30
18 19

⇒
20
15 25
10 18 30
16 19

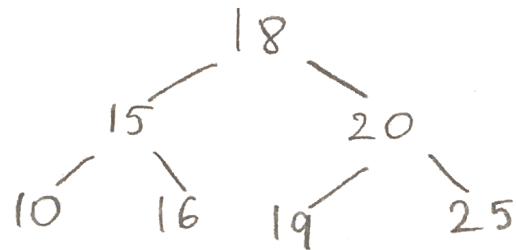
ii:



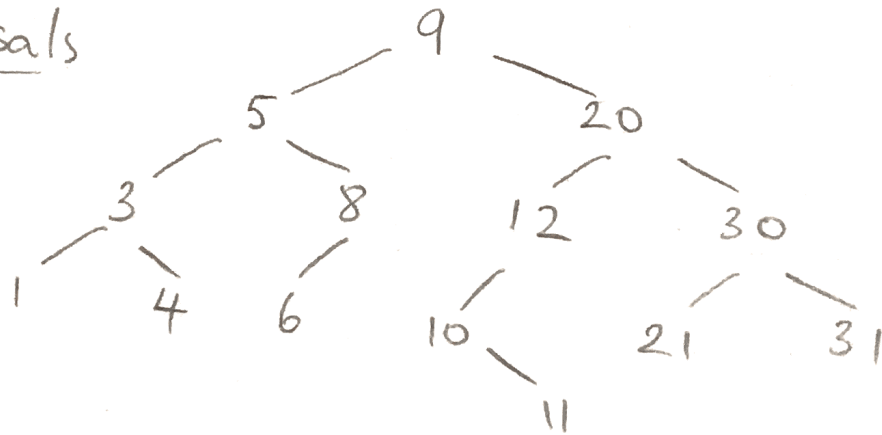
delete
30



\Rightarrow



3. Tree traversals



2.2