23.09.2024

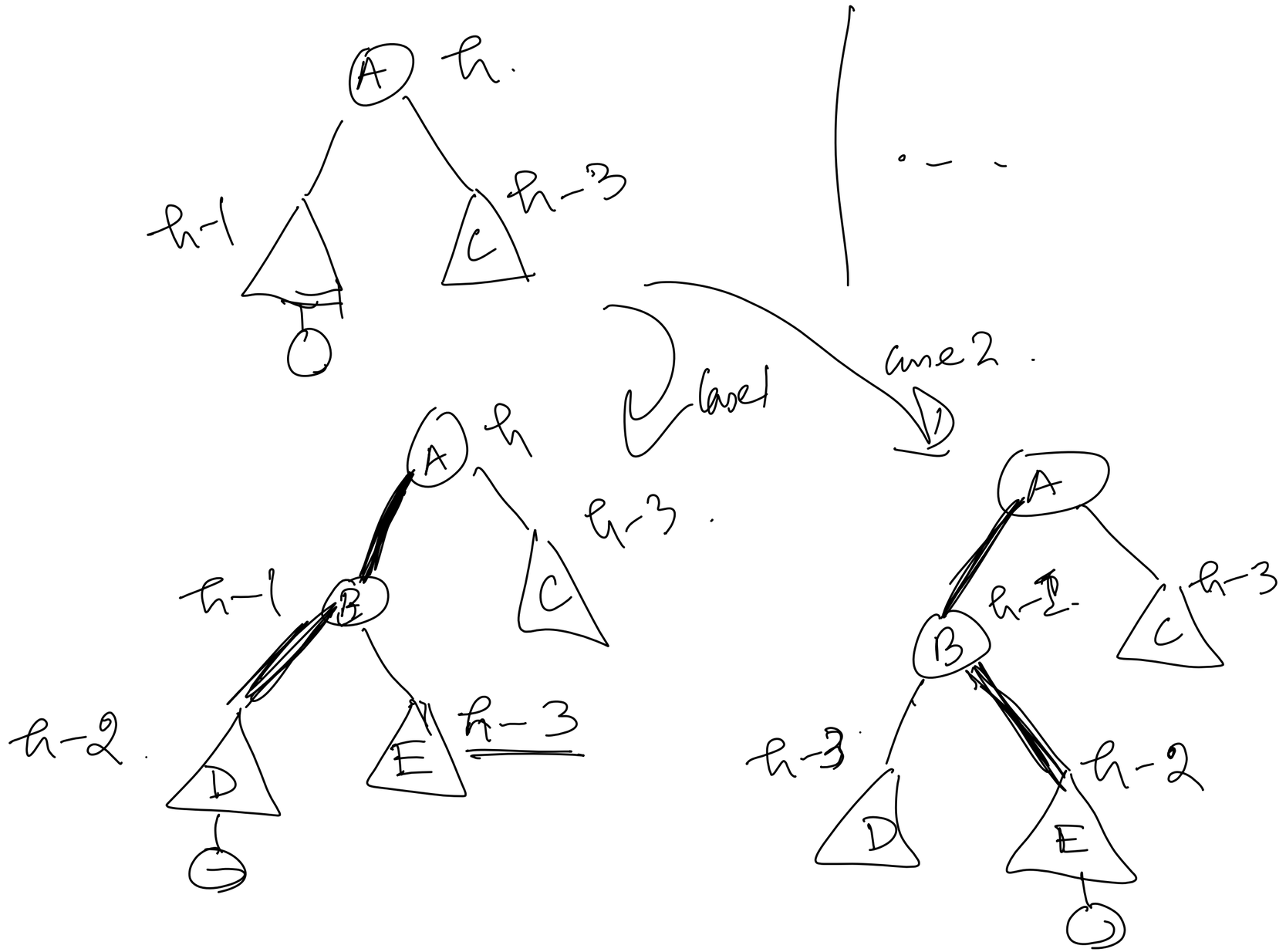
Recap : AVL Tree :

height(NULL) = 0

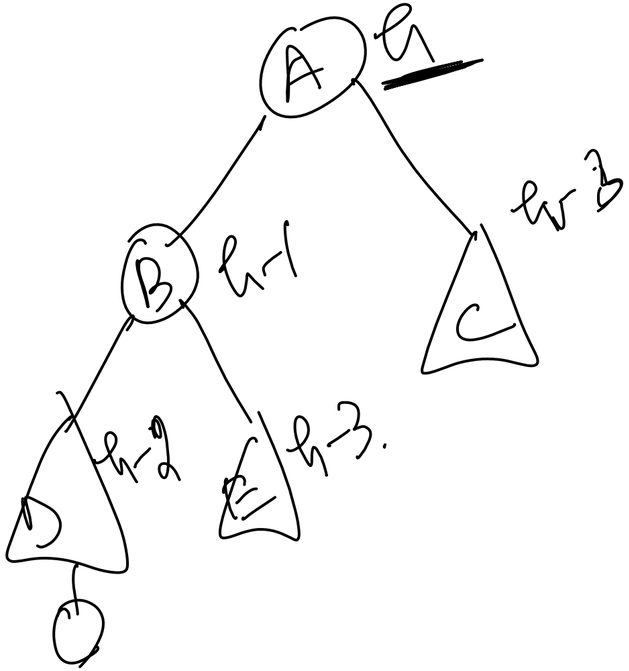
height(parent) = 1 + max { height(left-child), height(right-child) }

↳ $|ht(left\_child) - ht(right\_child)| \leq 1$

Insert :

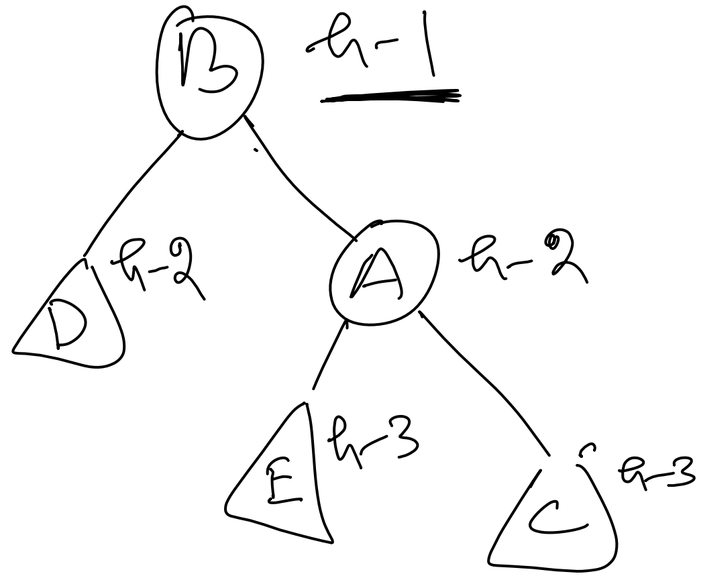

A
h.
h-1
h-3
C
...
case 2.
Level
A
h
h-3.
h-1
B
C
h-2.
D
E
h-3
A
h-1
B
h-3
C
h-3.
D
E
h-2

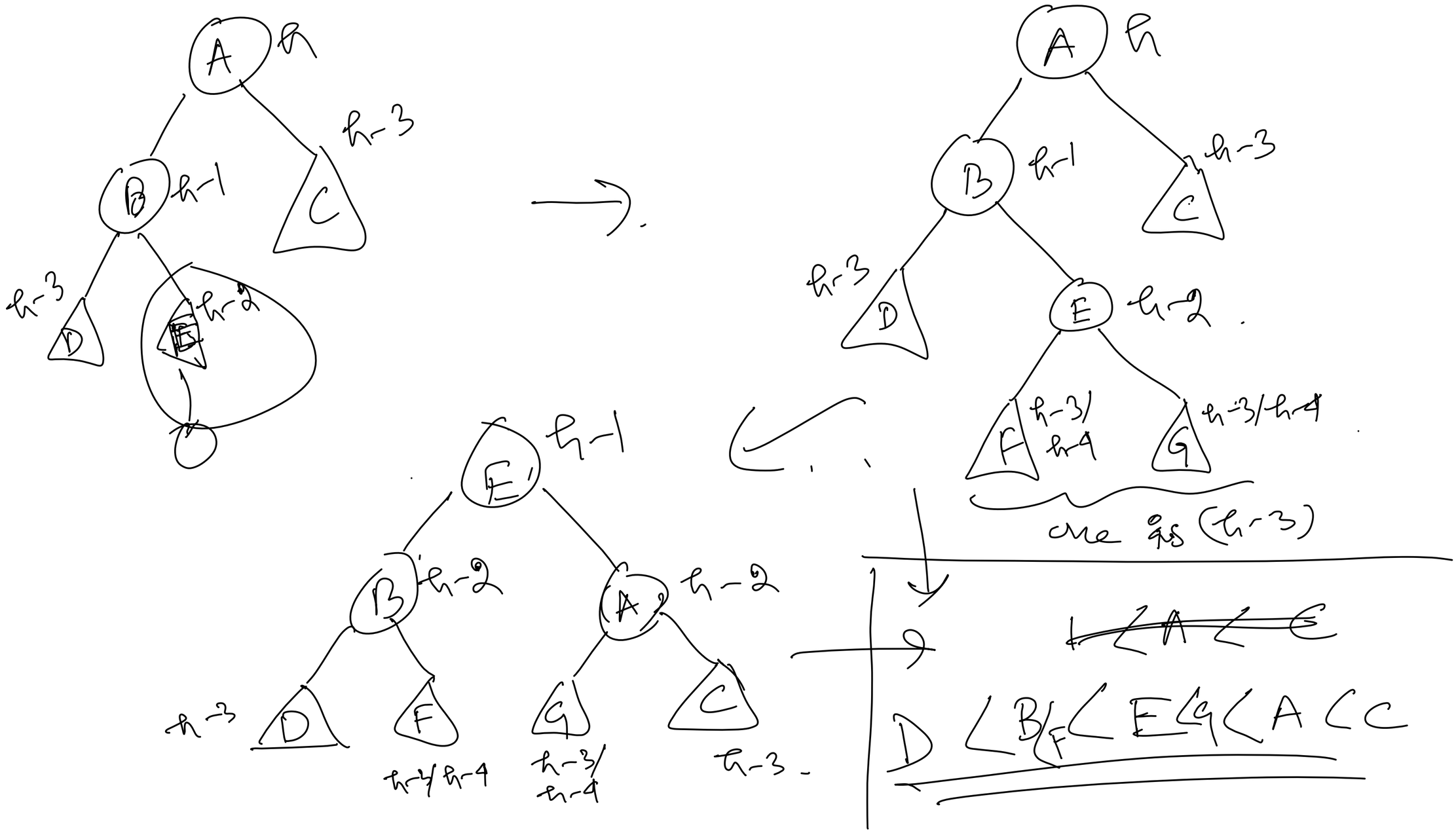
Restoring balance after insert



Single

no balance




A
h
B
h-1
C
h-3
D
h-3
E
h-2
A
h
B
h-1
C
h-3
D
h-3
E
h-2
F
h-3/
h-4
G
h-3/h-4
one as (h-3)
E
h-1
B
h-2
A
h-2
D
h-3
F
h-3/h-4
G
h-3/
h-4
C
h-3.
D
B
F
E
G
A
C

Exercise

- handle mirror images
- ponder over the pseudocode.
- check that relative ordering is preserved.

$O(1) + O(h) = O(h)$

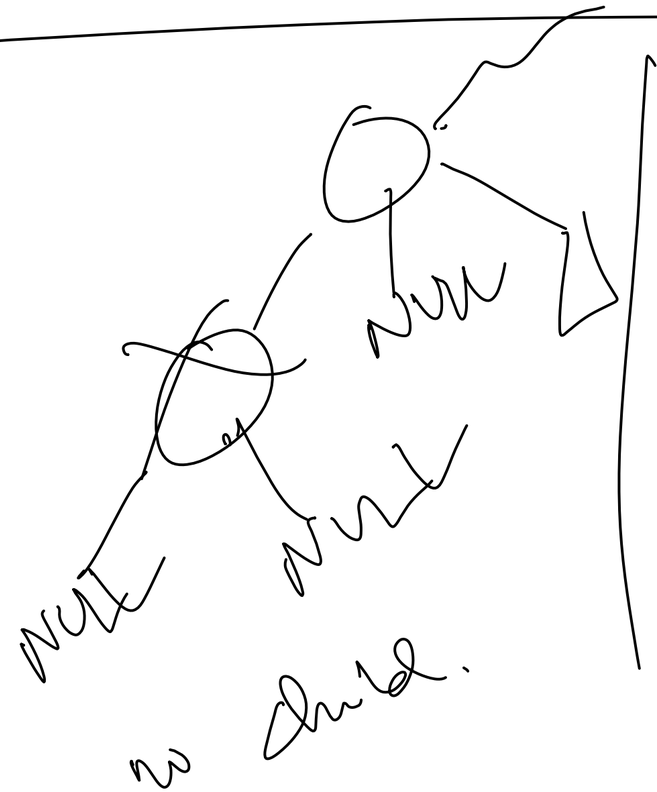
↓ height of tree.

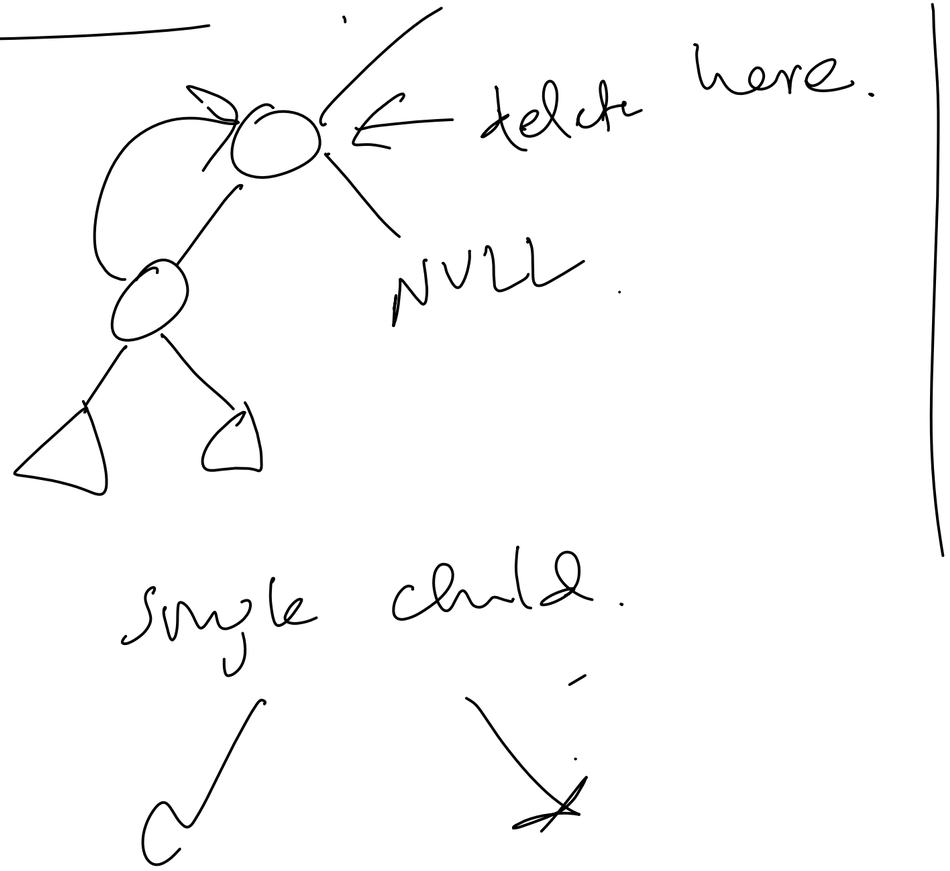
Sorting algorithm: n numbers.

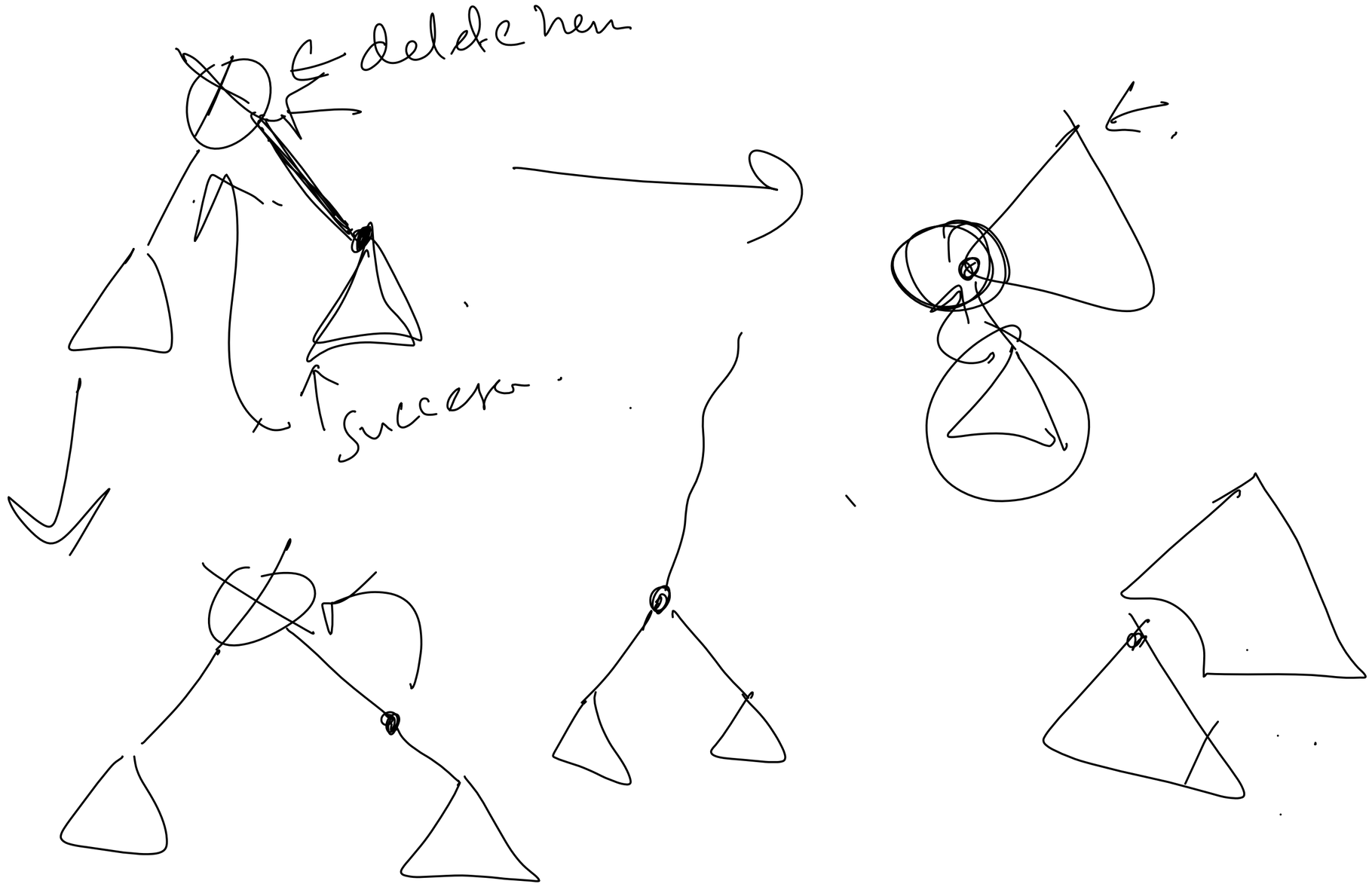
AVLSort: $\underbrace{n \cdot O(h)}_{\text{insertion}} + O(n) = O(h\,n)$

↓ $\log n$.

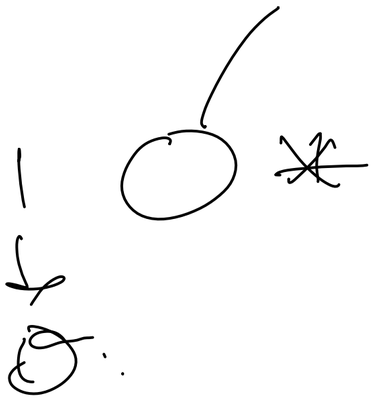
Restoring balance after delete






delete here
successor

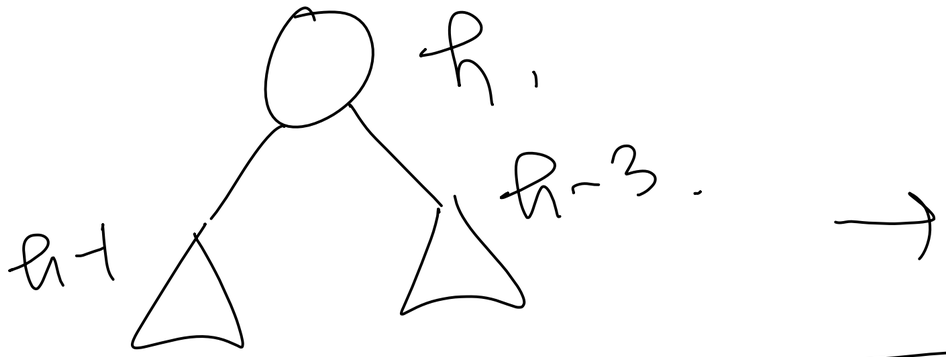
The deepest location where there is a problem has either no children or single child.

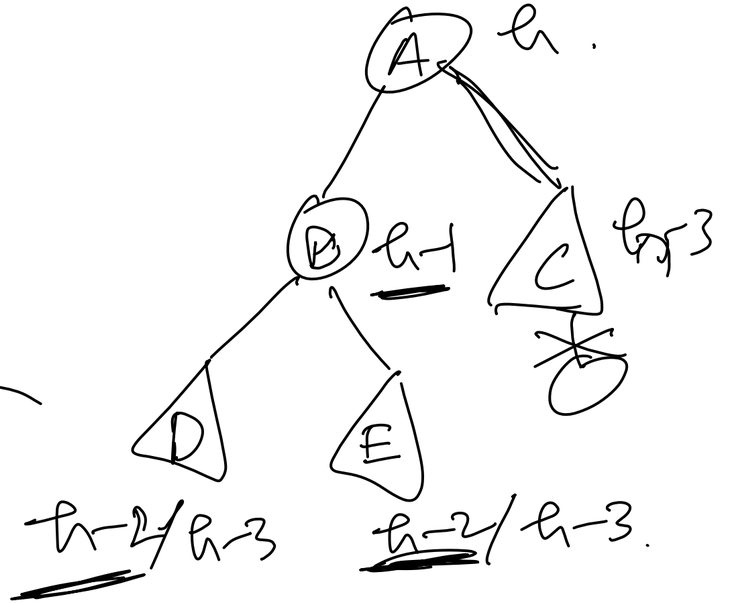


can

← (location of first problem is either a leaf or parent of a leaf)

we'll again go up & restore.



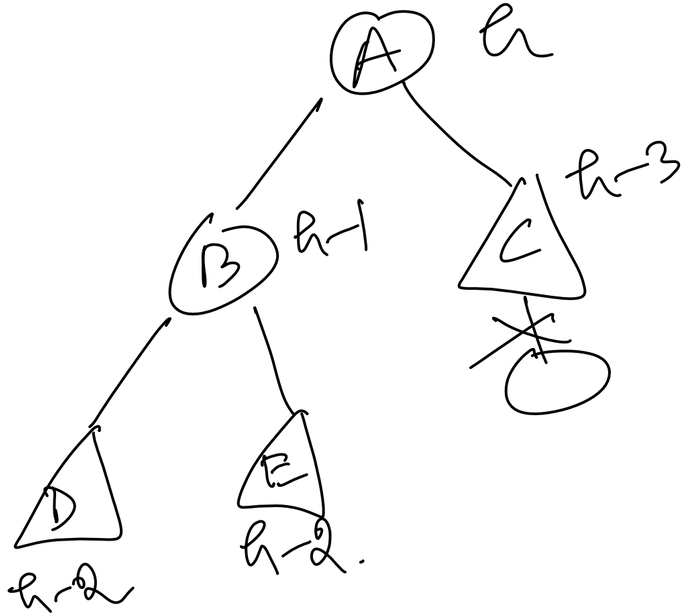


Case 3.
next page.

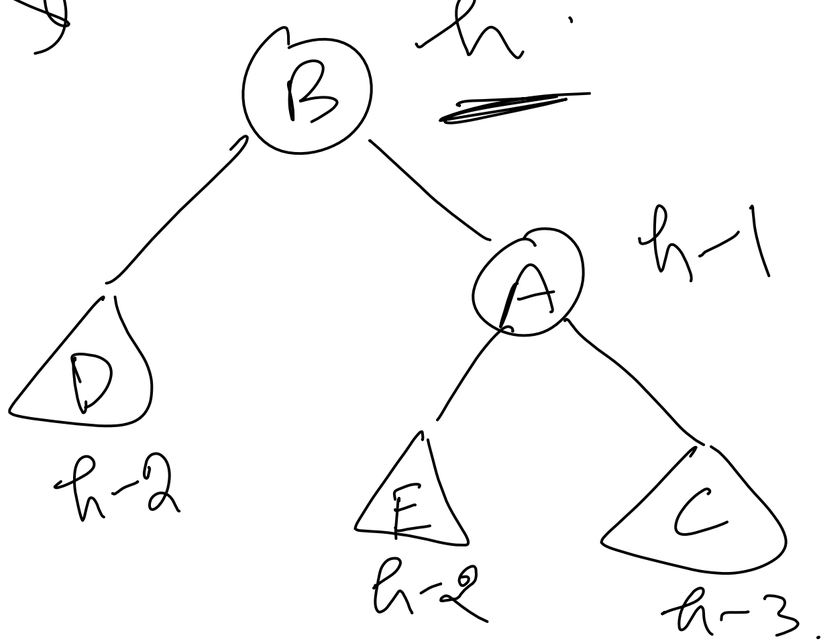
Case 1 as before.

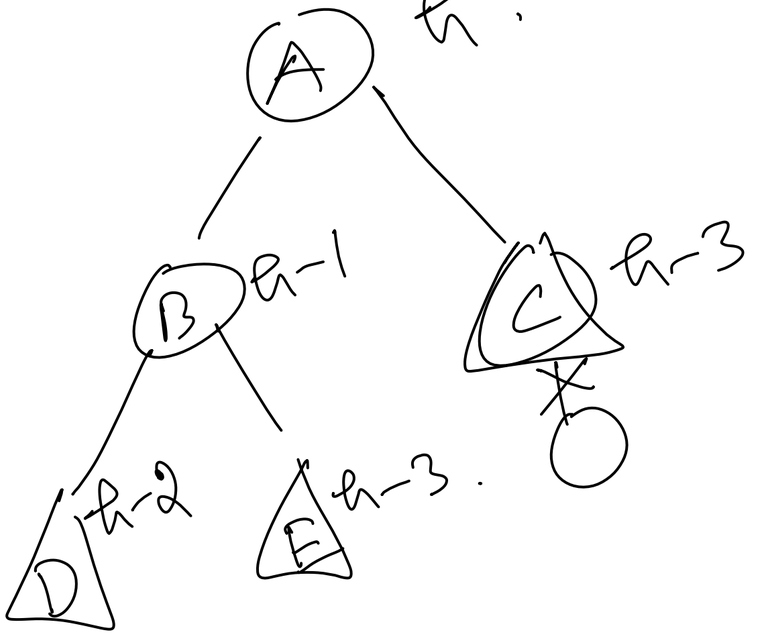
Case 2 as below.

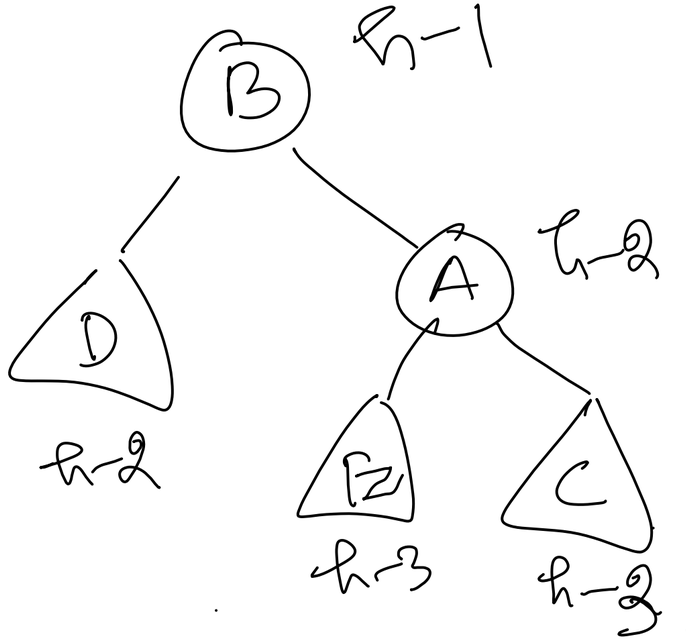
Case 3



single rotation.







In deletion, height of the ~~tree~~ subtree is going down.

- traverse up the tree & restrore if needed : while ( )

Time Complexity : Delete + rotations

O(1) O(h)

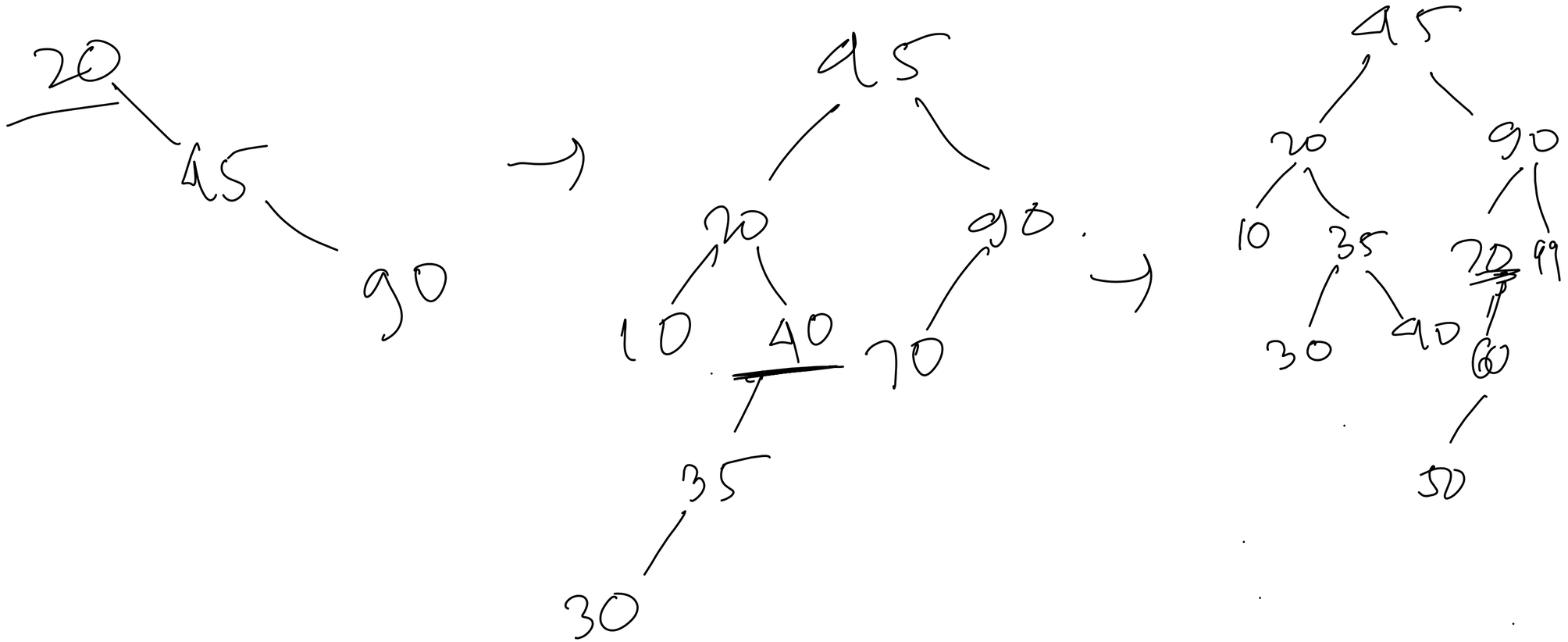
O(1).

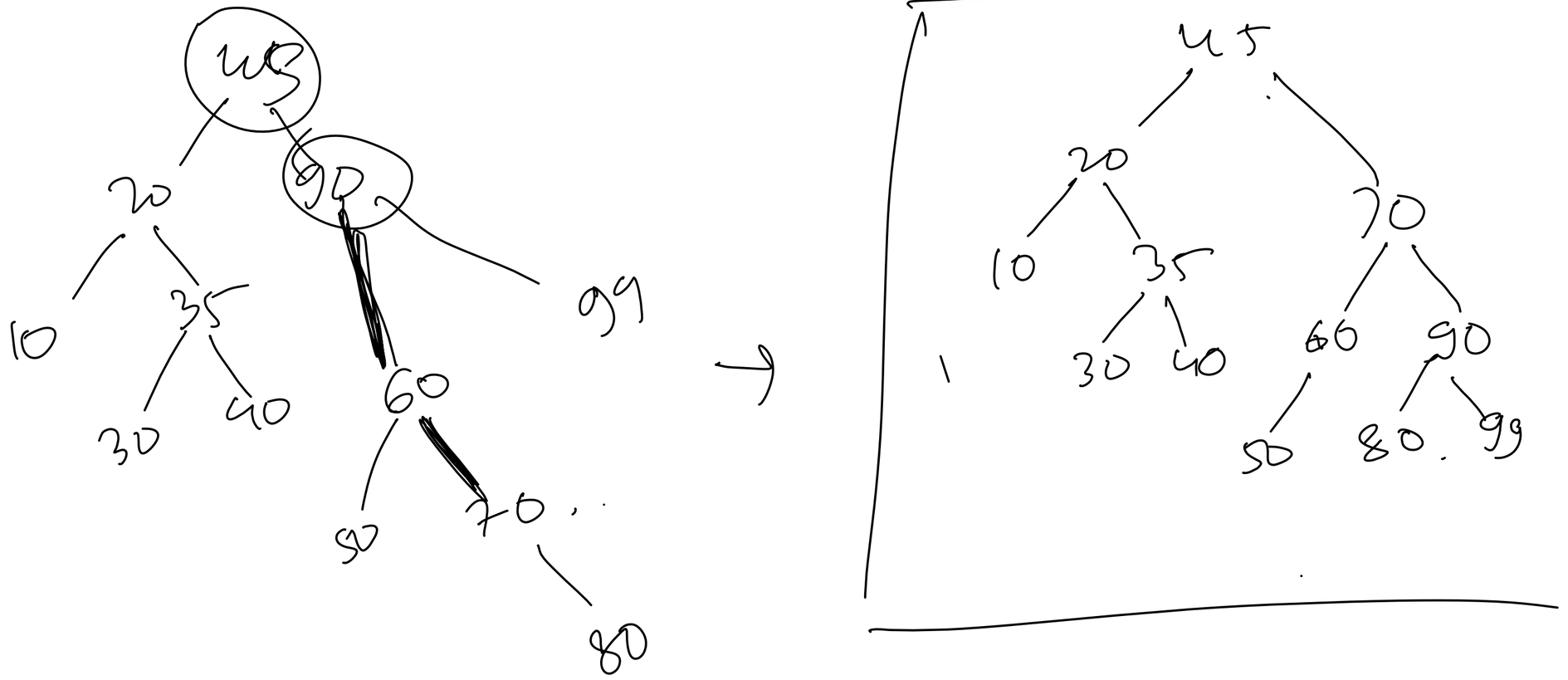
= O(h).

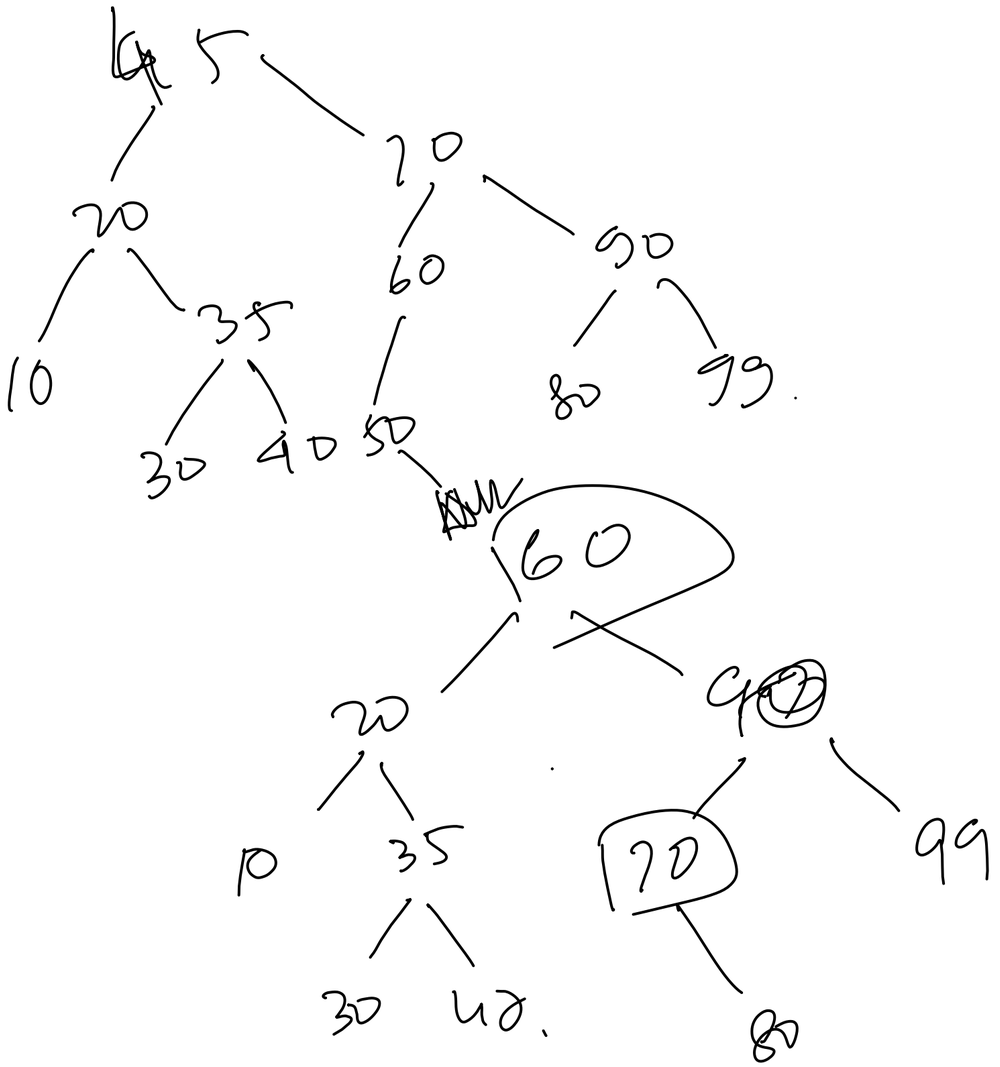
↓

height of tree.

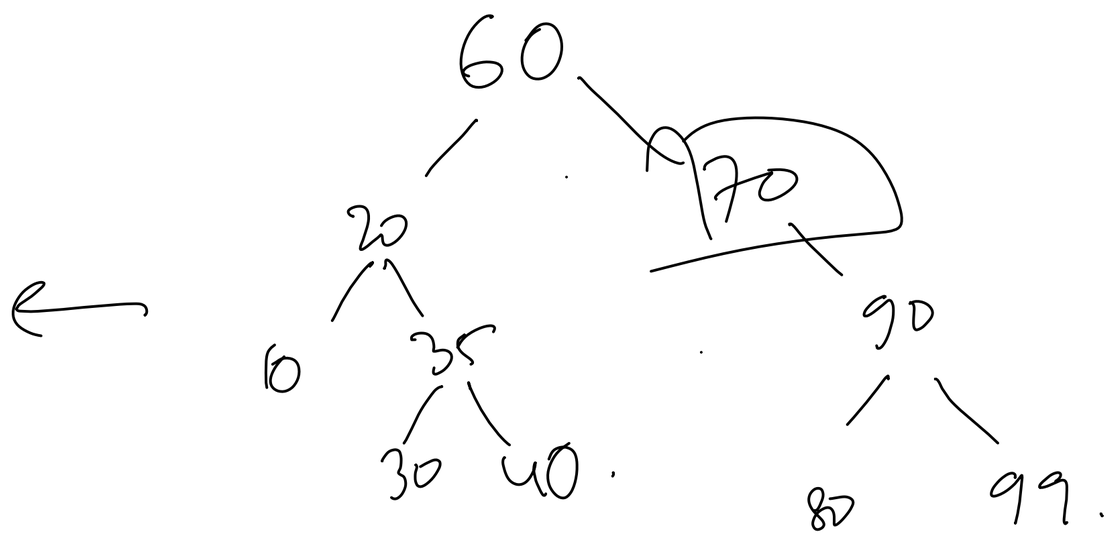
Example: 20, 45, 90, 70, 10, 40, 35,
30, 99, 60, 50, 80.

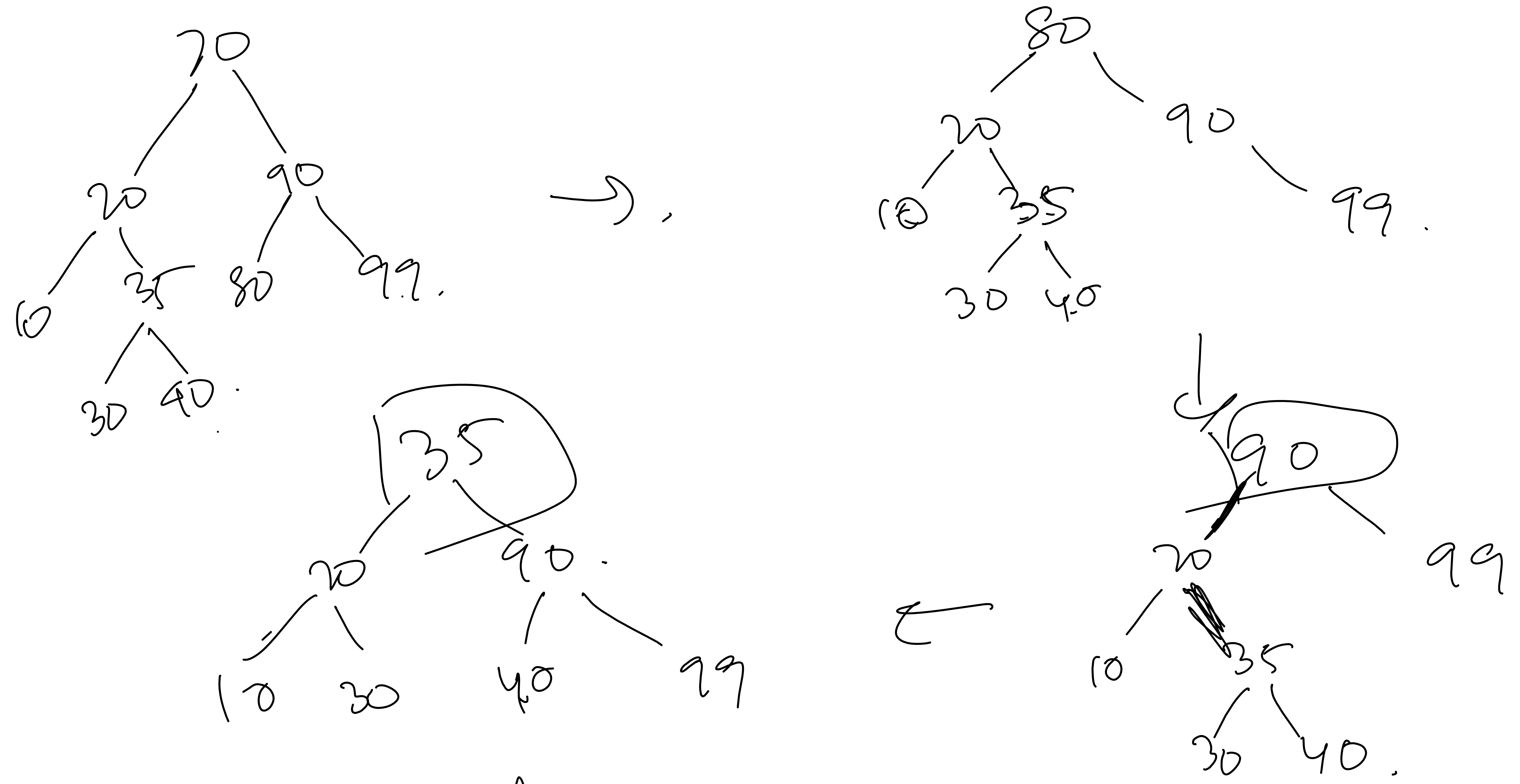



45
20
90
10
35
99
30
40
60
50
70
80
45
20
70
10
35
30
40
60
90
50
80
99


50
20
70
10
35
60
90
30
40.
80
99.

60
20
40
10
35
70
99
30
40.
80


60
20
70
10
35
90
30
40.
80
99.



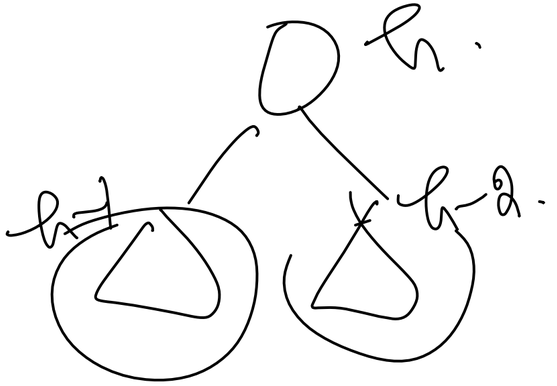
→ exercise

- delete: may need to go up

Claim: If $\exists$ n nodes, $h = O(\log n)$.

proof. If AVL has height h

$\Rightarrow \geq 2^{\Theta(h)}$ nodes.



$n(h) = ?$ min # nodes in height h AVL tree.

$n(1), \cdots, n(h-1)$

$n(h) \geq h(h-1) + n(h-2)$

$n(0) = 1$

$n(1) \quad 1$

$n(h) \geq n(h-1) + n(h-2) + 1$

$$n(q) \geq 2^{\Theta(q)}$$