A data structure

1. store data
2. support operations

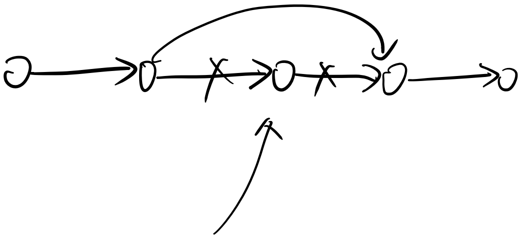
query
- find key
- find min
- find pre

update
- Insertion ...
- Deletion ...
- ...

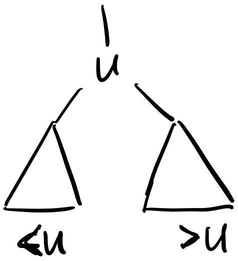
time

space

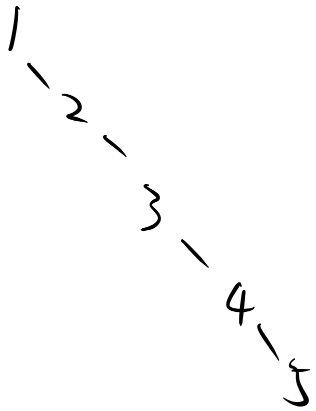
	linked list	binary search tree (BST)
find key	O(n)	O(h)
Insertion	O(1)	O(h)
Deletion	O(n)	O(h)

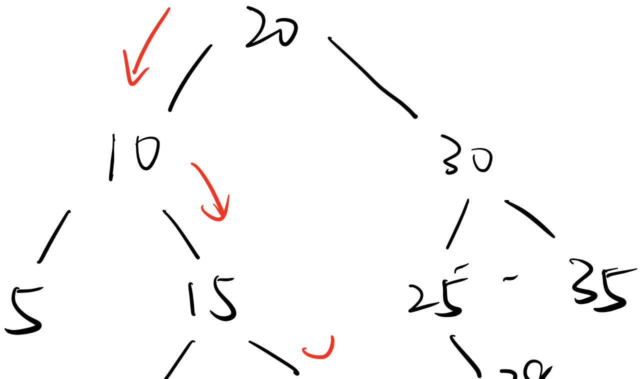


BST property



h=n



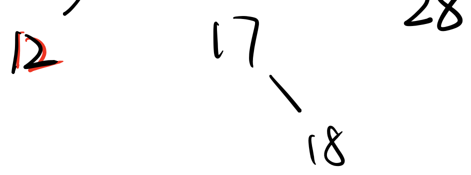


find key (16)

Ins (18)

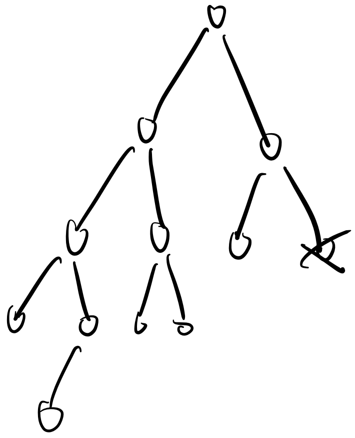
Del (12)

Del (17)



Del(20)

Idea 1 complete binary tree   Not work



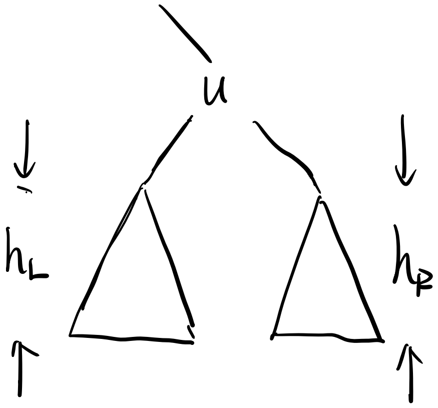
$h = O(\lg n)$

findkey $O(\lg n)$

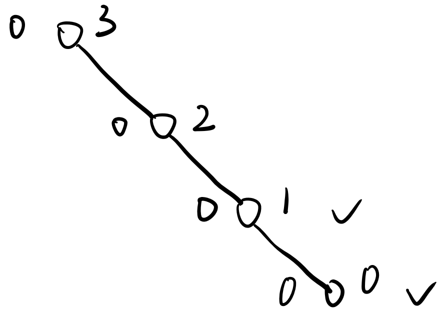
Ins $O(\lg n)$ + time to restore the balance $O(n)$

Del $O(\lg n)$

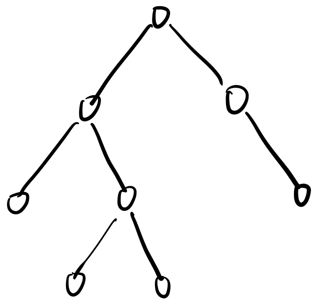
Idea 2



"balanced" if $\underbrace{|h_L - h_R|}_{\text{balance factor}} \le 1$

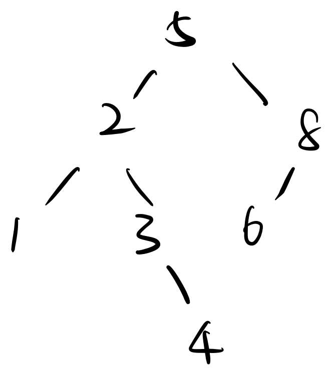


imbalanced tree



balanced tree

An AVL tree is a $\underset{②}{\underline{\text{balanced}}}$ $\underset{①}{\underline{\text{BST}}}$.



← AVL

Lemma

A balanced binary tree with n nodes must have height $O(\lg n)$

Proof:

Any balance binary tree of height h has at least c^h nodes.

c ↓ constant.

$$n \geq c^h \Rightarrow h \leq \lg_c n = O(\lg n)$$

$n(h)$: the smallest # nodes to make a balanced binary tree with height h (BBT)

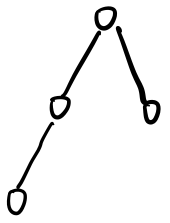
$n(0) = 0$

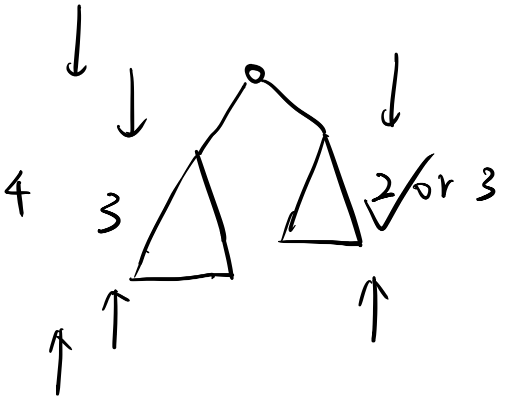
$n(1) = 1$

$n(2) = 2$

$n(3) = 4$

$n(4) = 1 + n(3) + n(2) = 7$

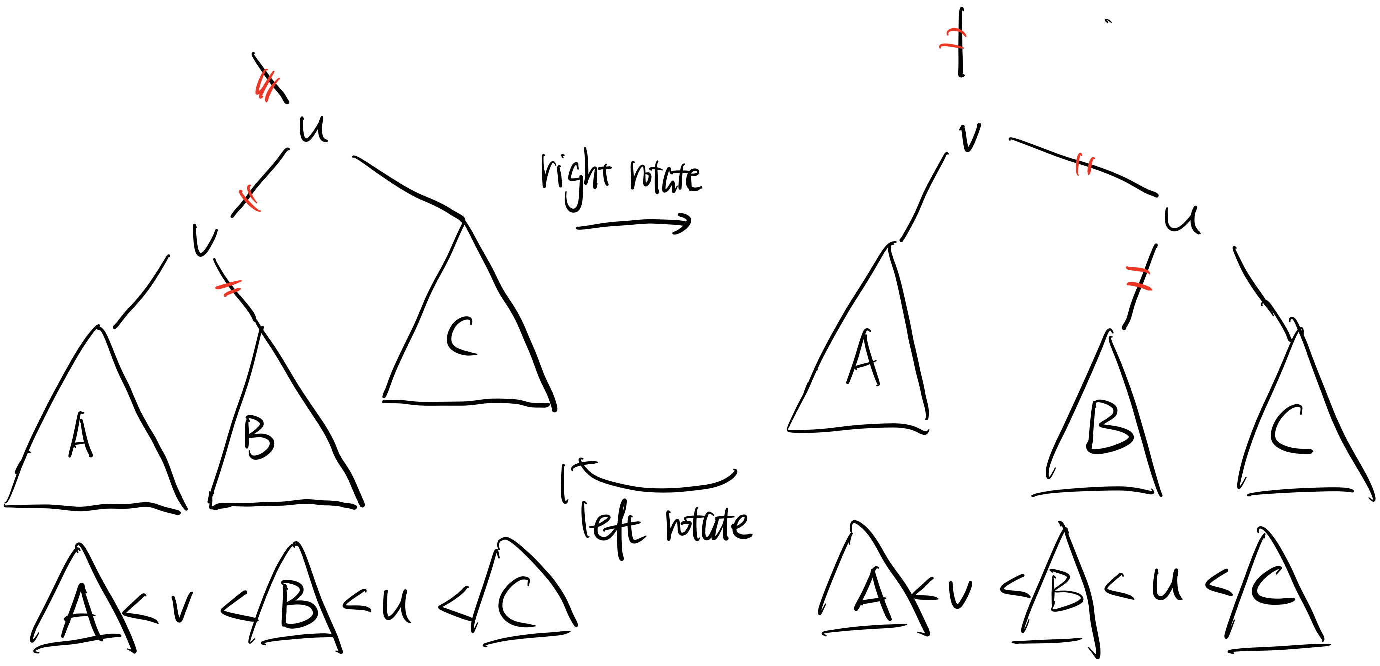




$$n(h) = 1 + n(h-1) + n(h-2)$$

$$n(h) \geq \left(\frac{1+\sqrt{5}}{2}\right)^h$$

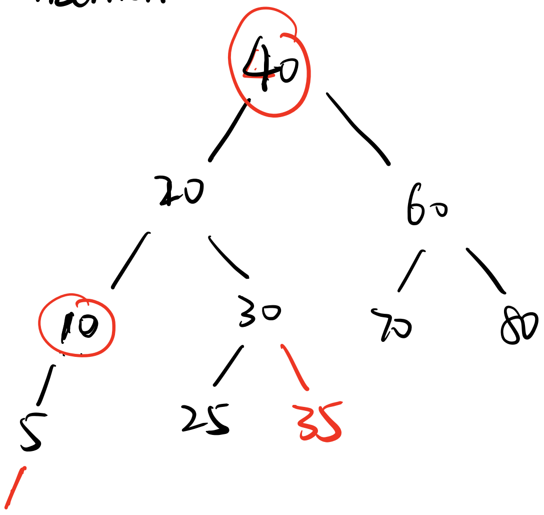
Rotation



preserve BST property

O(1) time

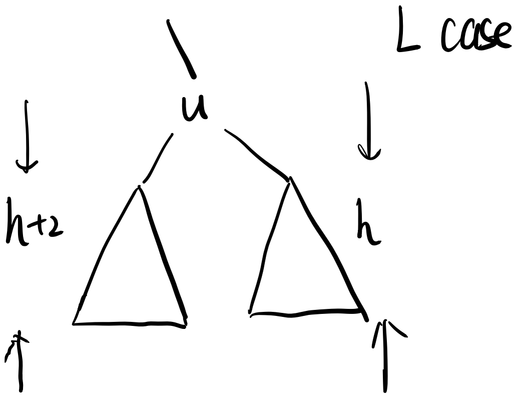
Insertion

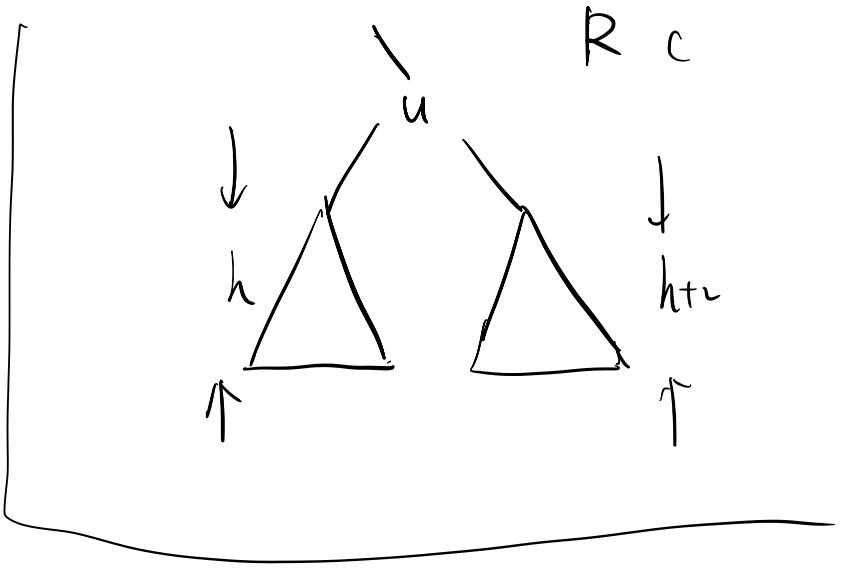


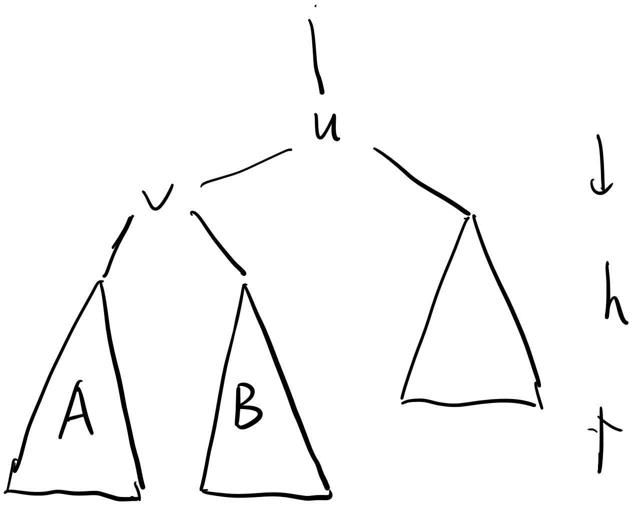
Ins(35)

Ins(2)

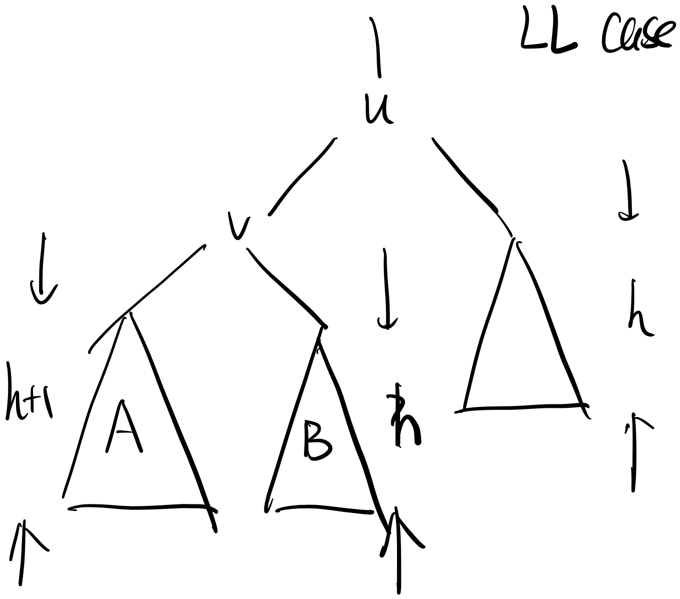
2

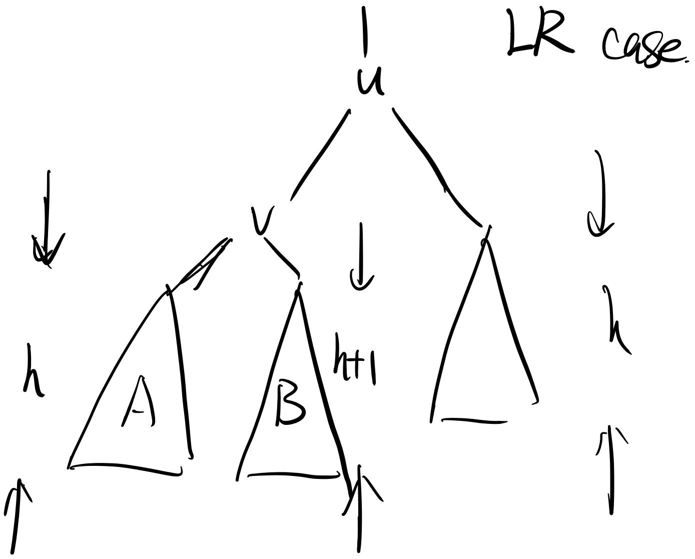






at least one of A and B have height h+1



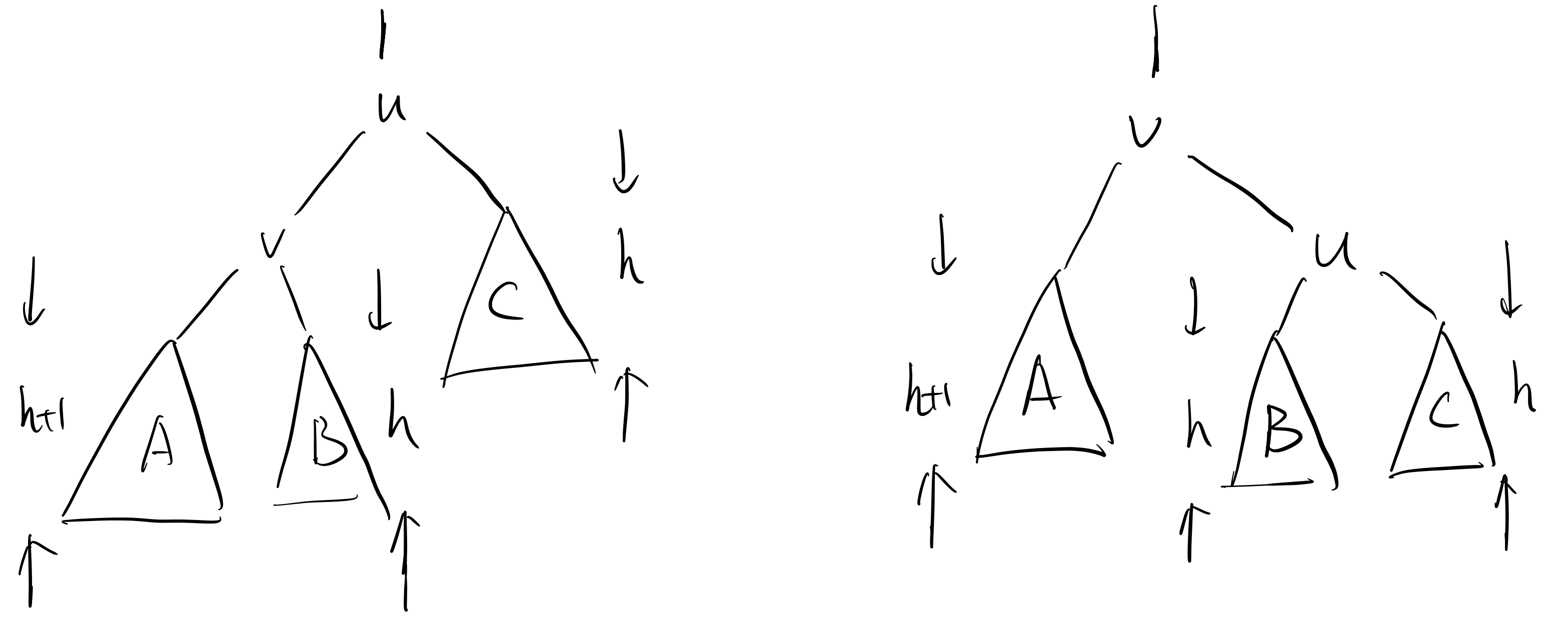


> h+1 ? ✗

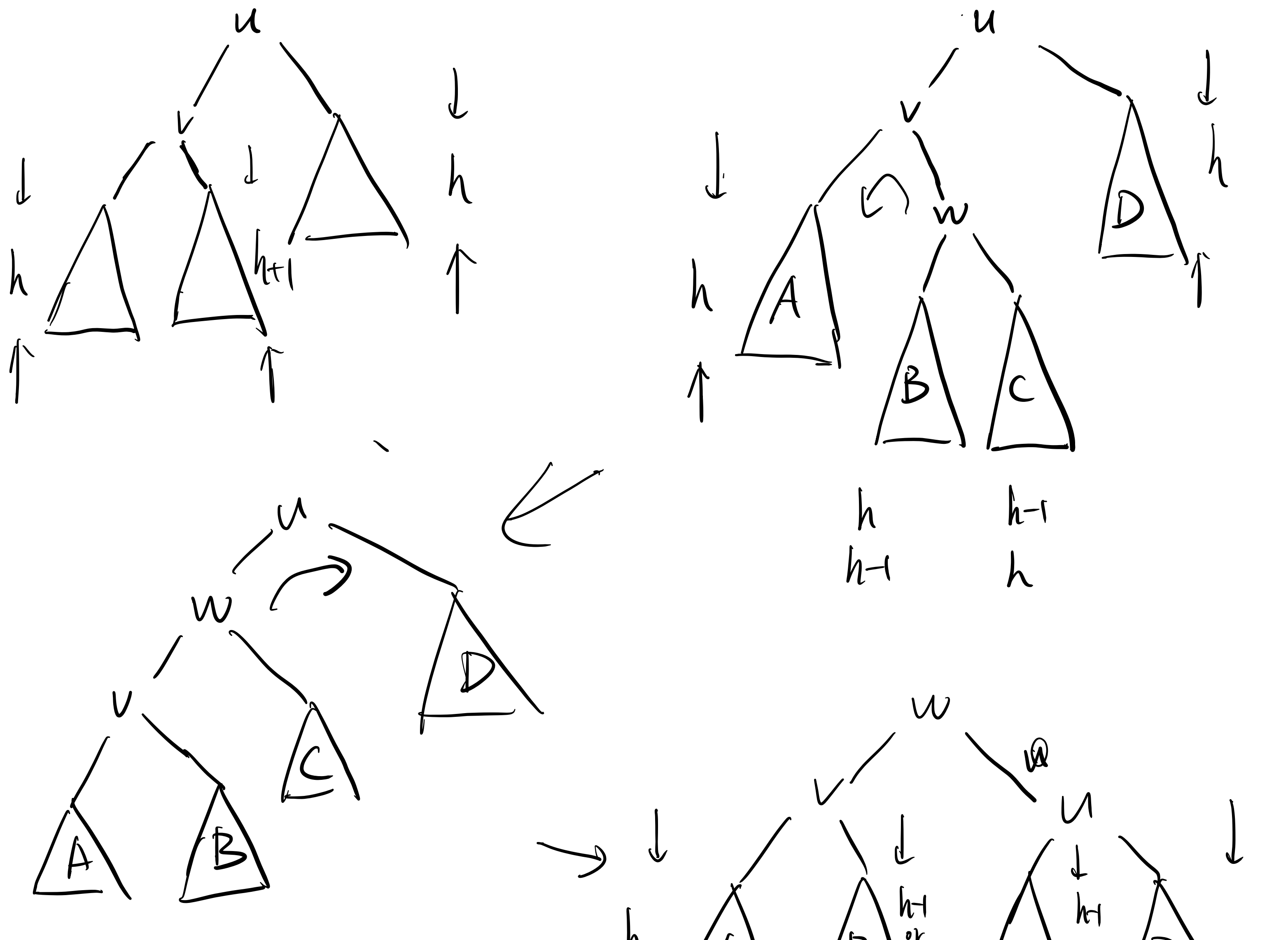
< h ?

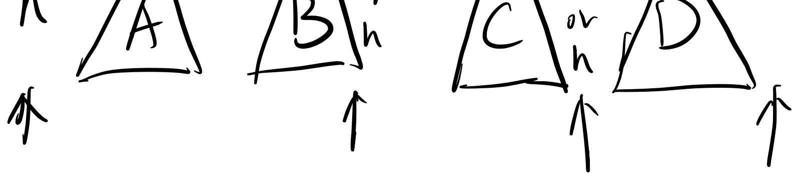
h or $\boxed{h+1}$? ✗

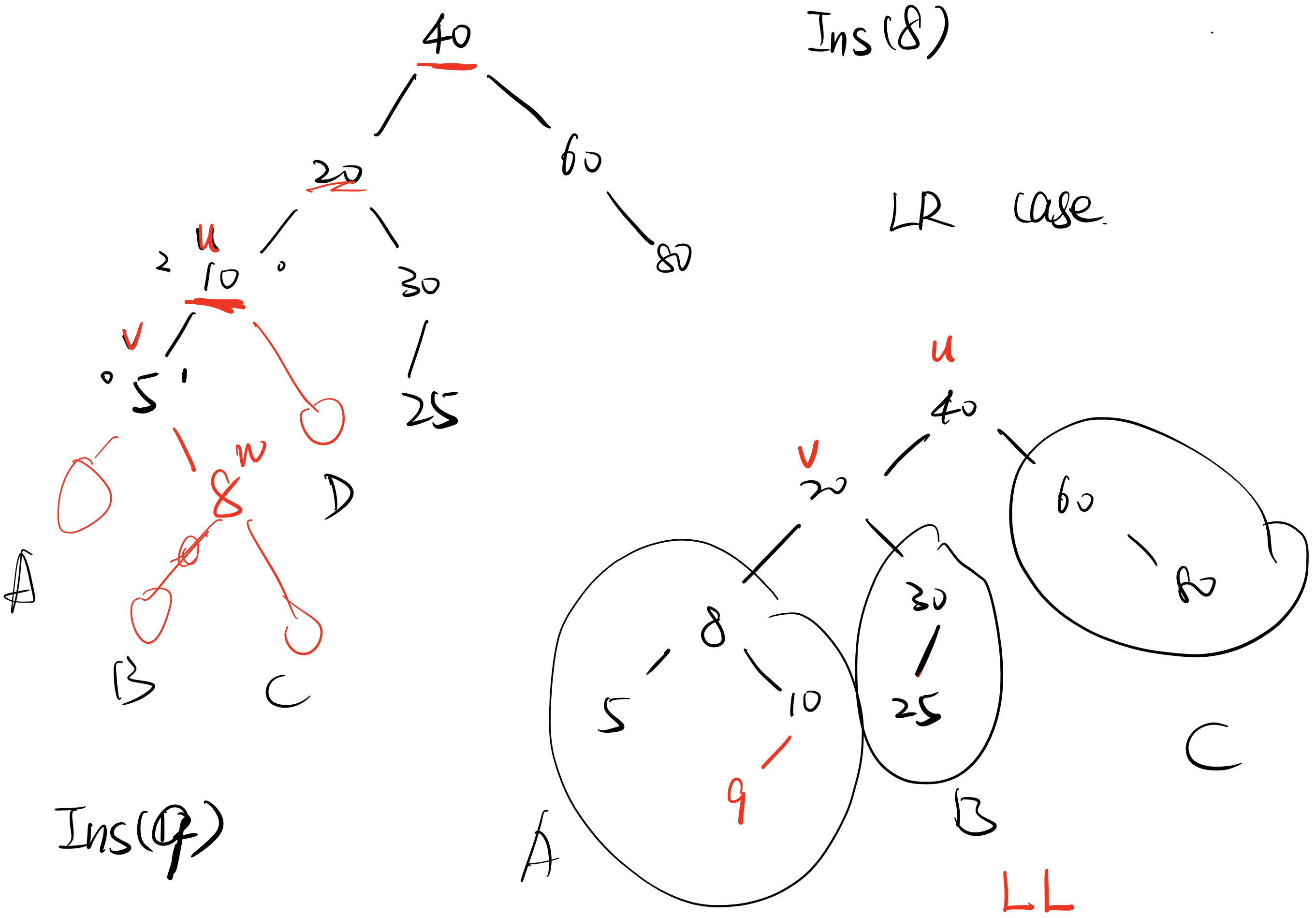
LL case

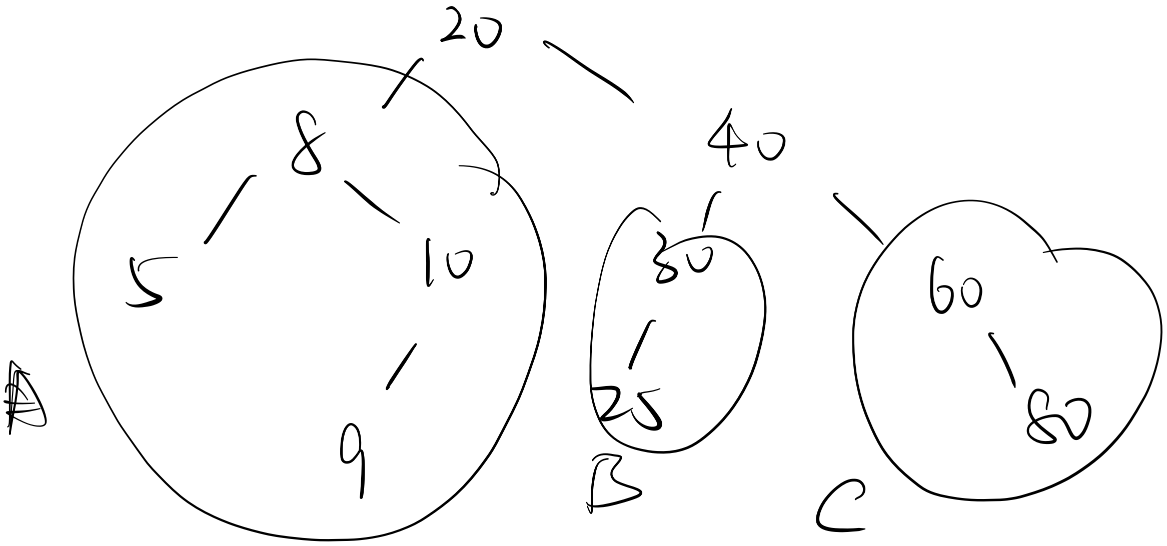


LR case

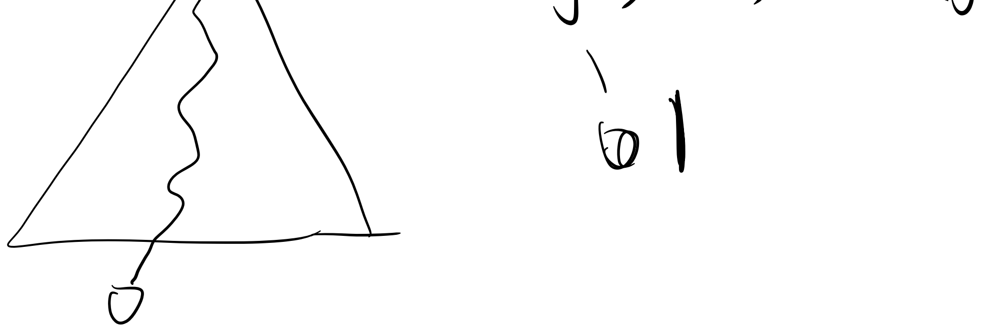








$\triangle^{O(\lg n)} + O(\lg n) \cdot O(1) = O(\lg n)$



Deletion

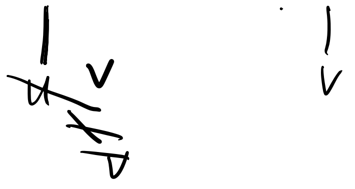
Every BST deletion in AVL tree is essentially removing a leaf.

delete(u)

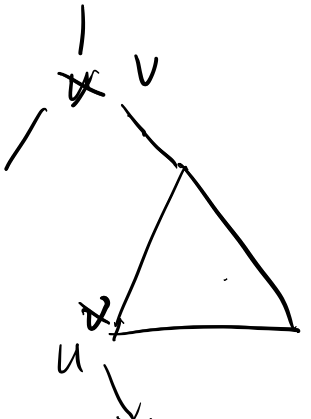
1. u is a leaf.

2. u has one child v.

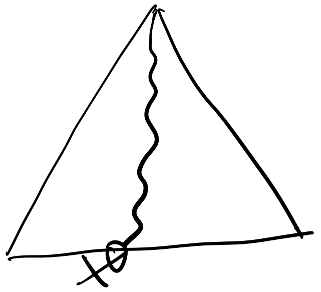
   v must be a leaf.

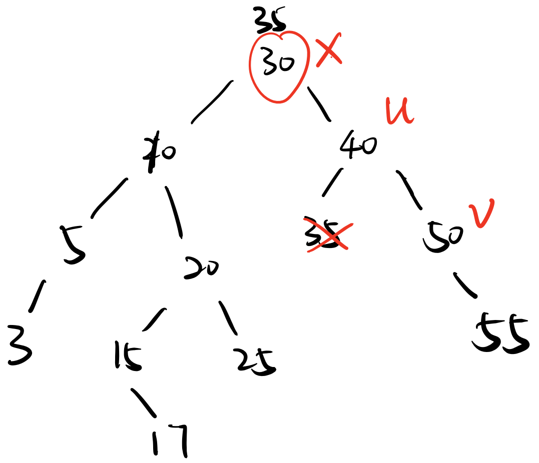


3. u has two children.



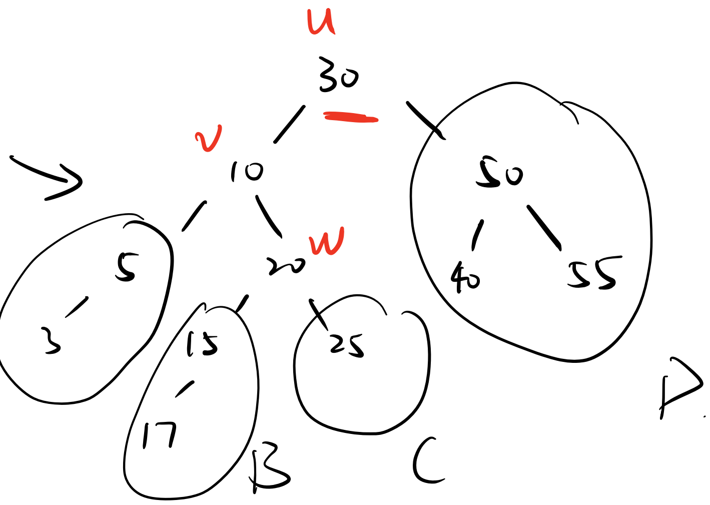
has at most one child case 1 & 2.

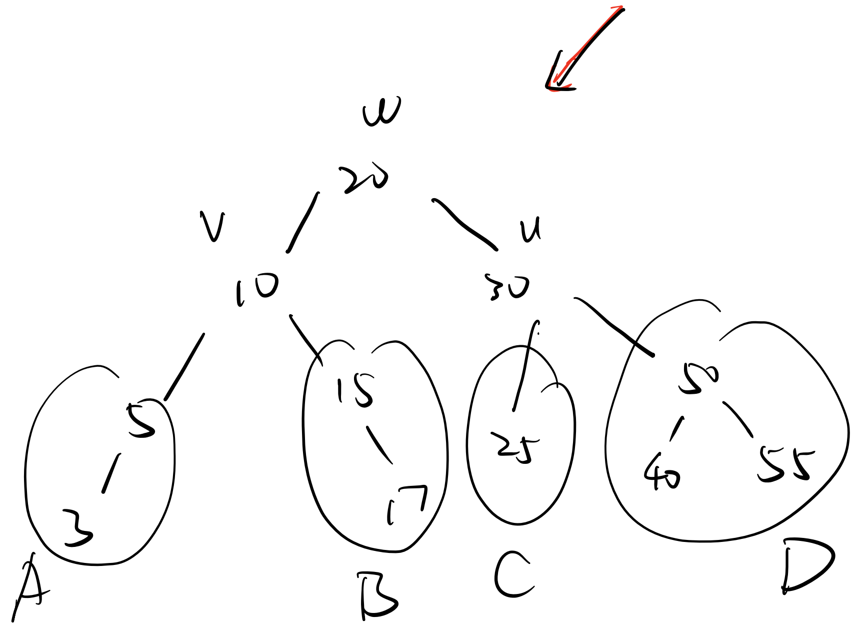




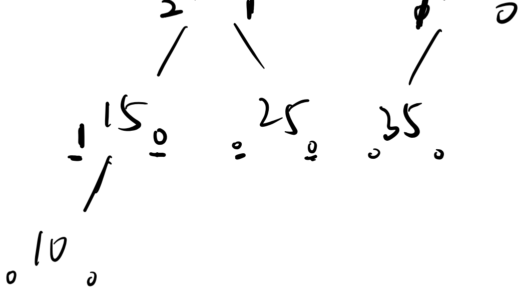
RR case.

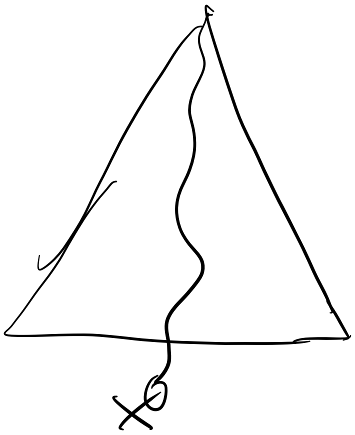
LR case.





deletion time $O(\lg n) + O(\lg n) \cdot O(1) = O(\lg n)$


2
1
0
0
1
15
0
0
25
0
0
35
0
0
10
0


×0