Szótárak:

init(t) (üresre init)

ins(t, k, ...)

has(t, k): ...

del(t, k)

rem(t, k): ...

min(t): ...

max(t): ...

remMin(t):

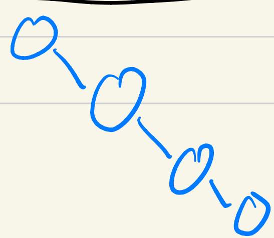
remMax(t):

Bin. kfák voltak

HASH TABLE

AVL fákon: $O(\log|t|)$ időben

max. $\Theta(\log|t|)$ időben



rossz esetben

szept. 12.

(magasság szerint)

D AVL fa: kiegyensúlyozott BST
(balanced)

D $(*p)$ csúcs egyensúlya: $p \to b \overset{def}{=} h(p \to right) - h(p \to left)$

$(*p)$ kiegyensúlyozott $\overset{def}{\iff} p \to b \in \{-1, 0, 1\}$

t bin. fa kiegyensúlyozott $\overset{def}{\iff} \forall$ csúcsa kiegy.

T (kiegyensúlyozott bin. fa magassága)

t kiegy. bin. fa; $n := n(t)$; $h = h(t)$

$\Rightarrow \lfloor \log n \rfloor \le h \le 1.45 \log n$

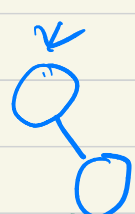
Biz. alapgondolata:

FIBONACCI
FAK

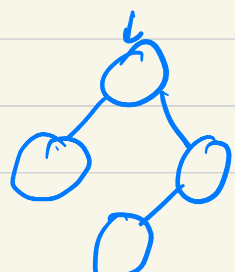
$f_h :=$ h magassághoz tartozó legkisebb kiegyensúlyozott bin. fa méret

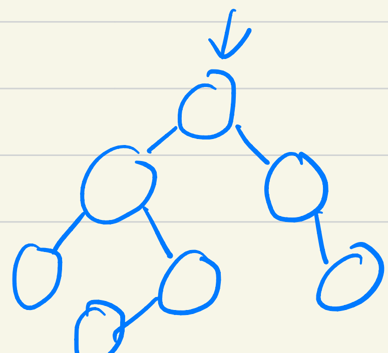
$f_0 = 1$

$f_1 = 2$



$h \geq 2 \Rightarrow 1 + f_{h-1} + f_{h-2} = f_h$

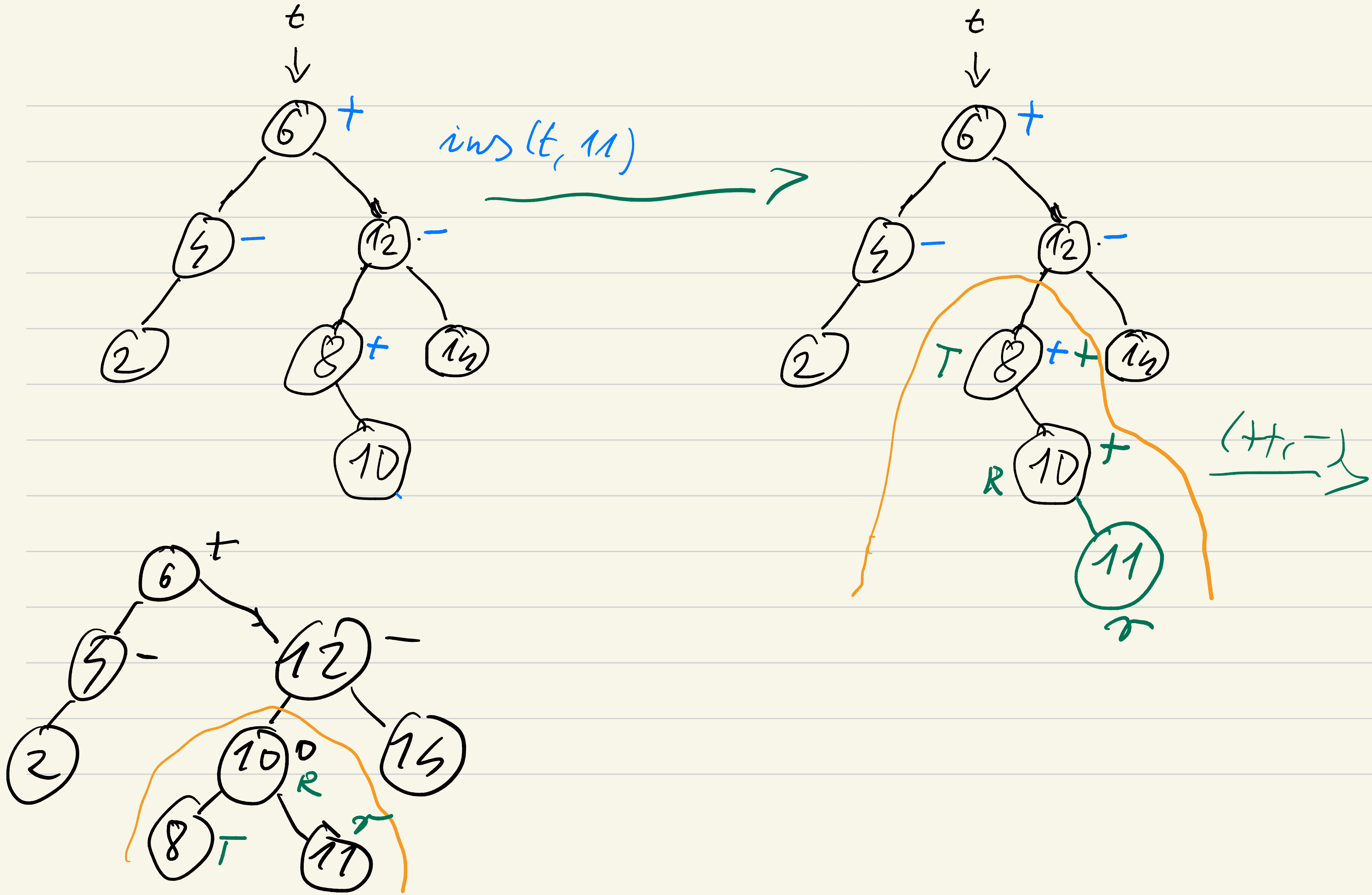




Tetsz. h mag. kiegy. bin. fára: $n \geq f_h$

Szótár (AVL fa) műveletek:

- $t := \emptyset$
- ins(t, k)
- search(t, k): $\mathbb{B}$
- min(t): Node*
- max(t) -||-
- remMin(t) -||-
- remMax() -||-
- del(t, k)

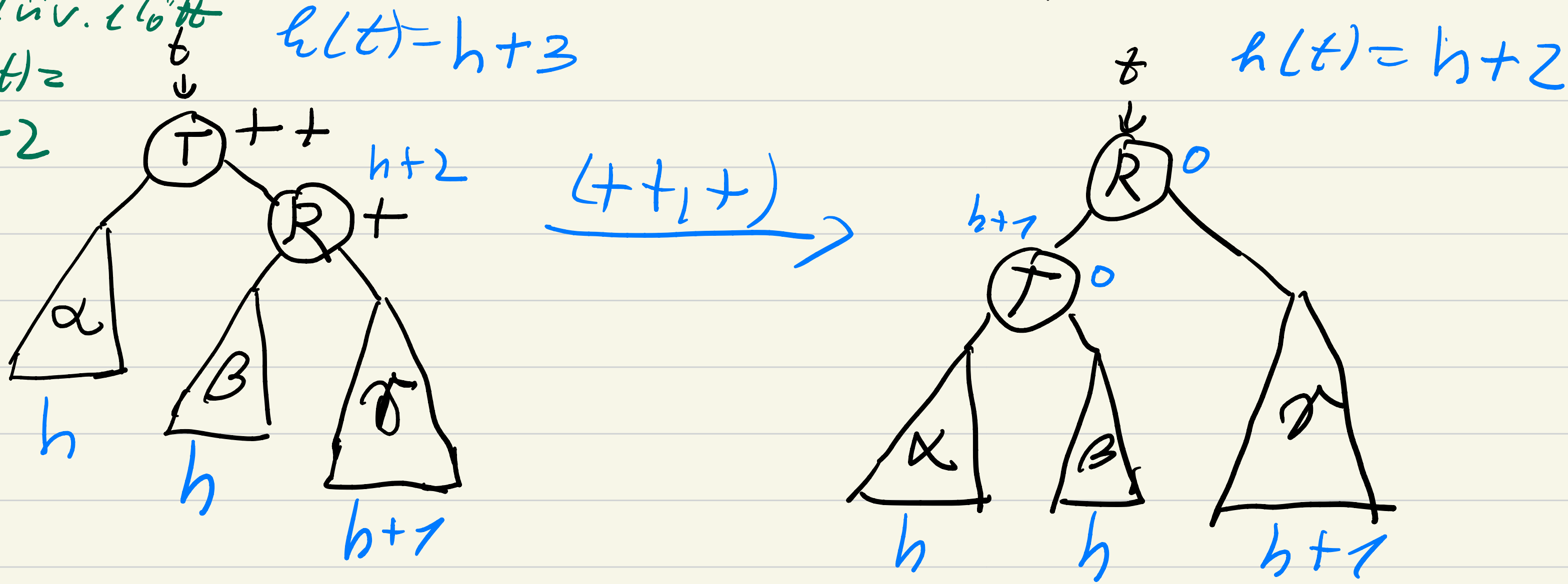


t
ins(t, 11)
t
(++, -)

Műv. előtt

$h(t)=$

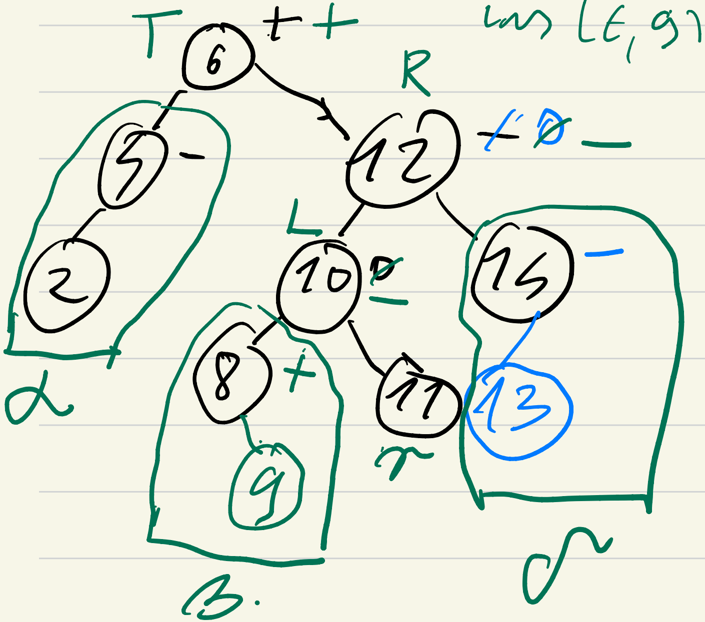
$h+2$



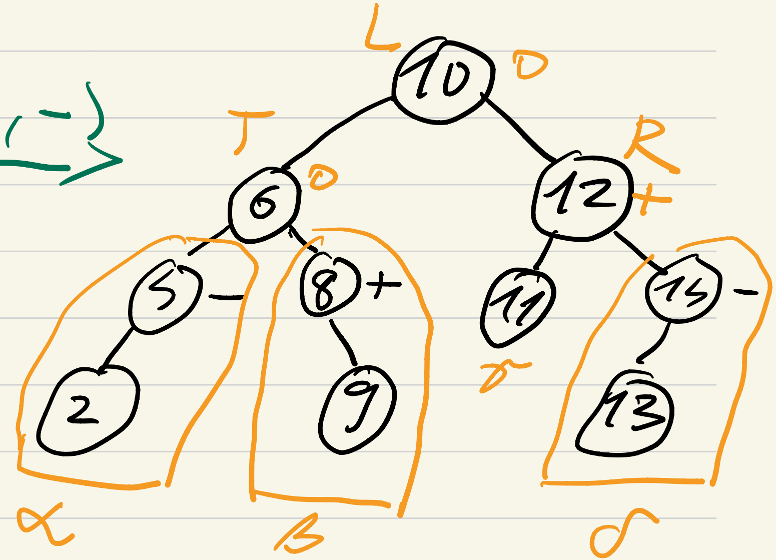
$\alpha < T < \beta < R < \gamma$

ins(t, 13)

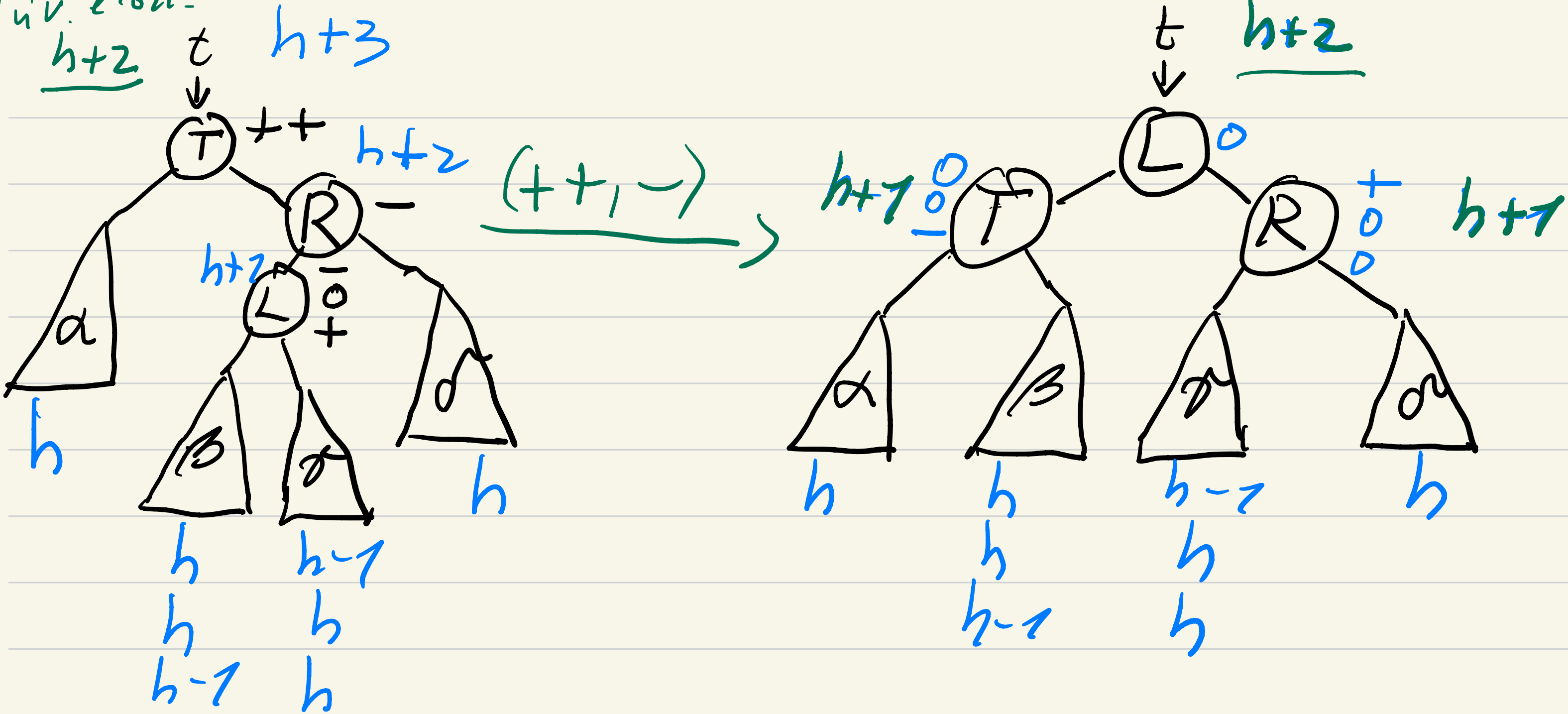
ins(t, 9)



$(++, -, -) \longrightarrow$

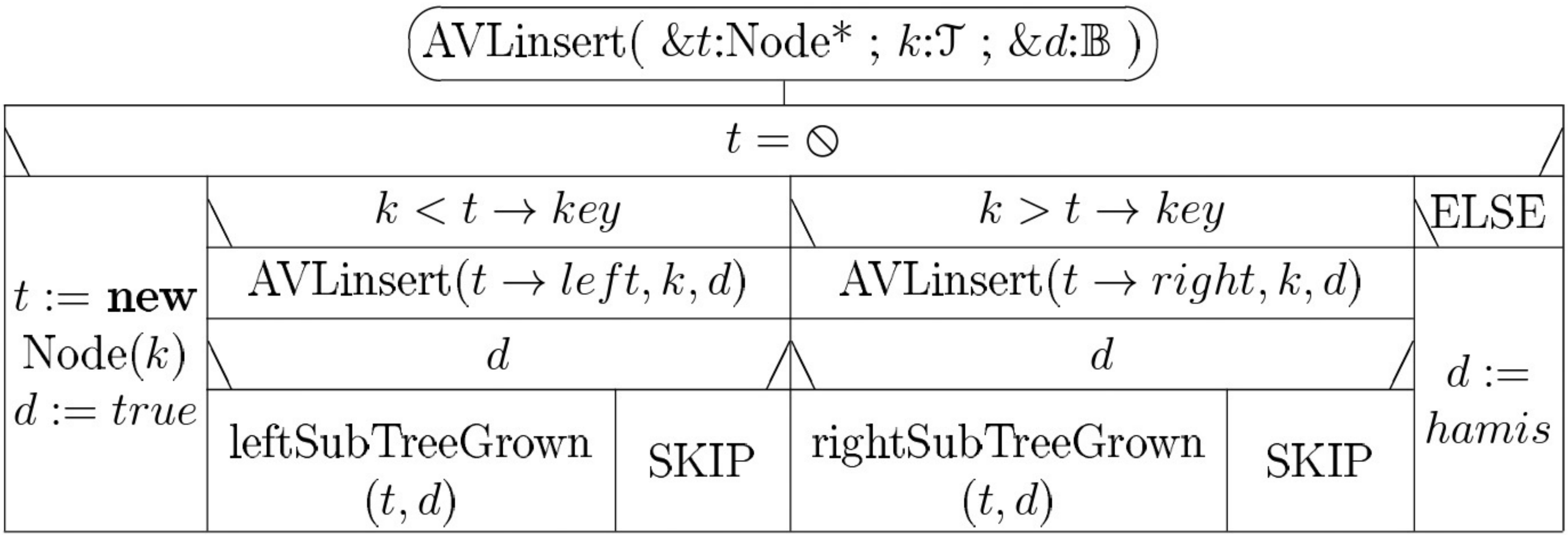


Műv. előtt:



$\alpha < T < \beta < L < \gamma < R < \delta$

Node
+ $key : \mathcal{T}$ // $\mathcal{T}$ is some known type + $b : -1..1$ // the balance of the node + $left, right$: Node*
+ Node() { $left := right := \oslash$; $b := 0$ } // create a tree of a single node + Node(x:$\mathcal{T}$) { $left := right := \oslash$; $b := 0$; $key := x$ }



leftSubTreeGrown(&t:Node* ; &d:$\mathbb{B}$)

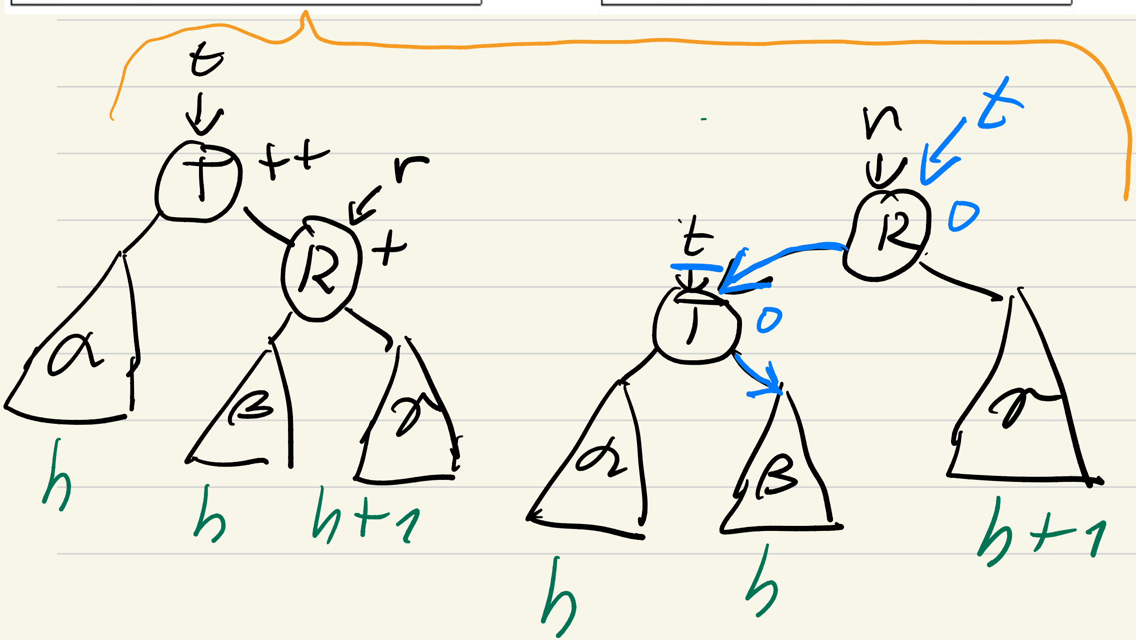
- **if** $t \rightarrow b = -1$
  - $l := t \rightarrow left$
  - **if** $l \rightarrow b = -1$
    - balanceMMm(t, l)
  - **else**
    - balanceMMp(t, l)
  - $d := false$
- **else**
  - $t \rightarrow b := t \rightarrow b - 1$
  - $d := (t \rightarrow b < 0)$

rightSubTreeGrown(&t:Node* ; &d:$\mathbb{B}$)

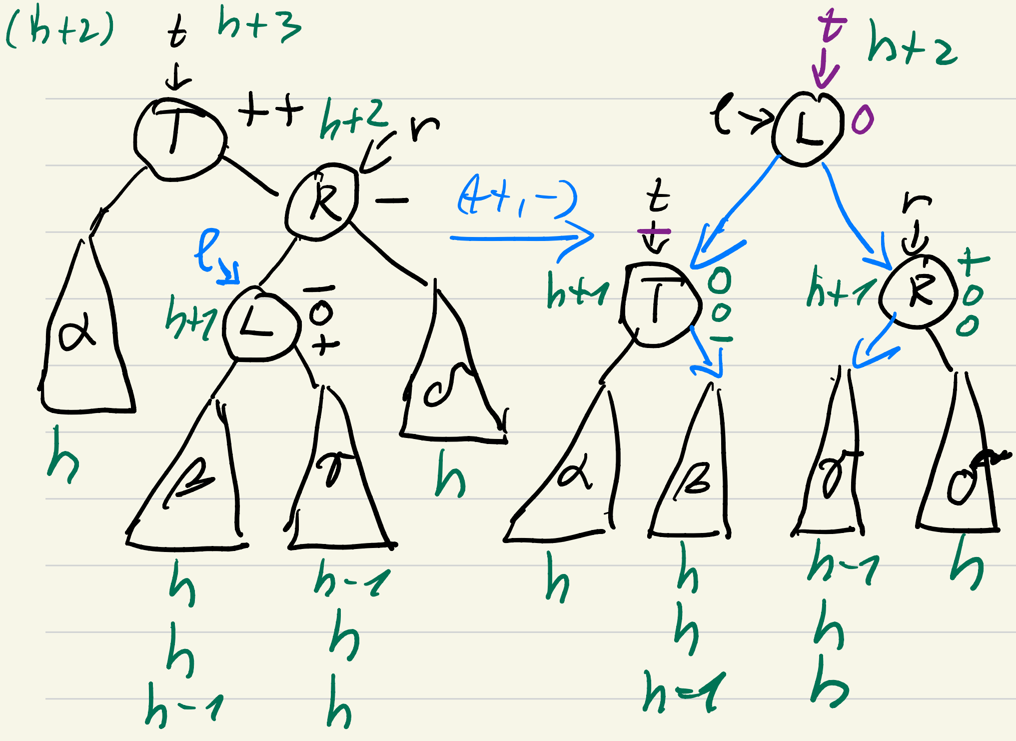
- **if** $t \rightarrow b = 1$
  - $r := t \rightarrow right$ //t->b:=2
  - **if** $r \rightarrow b = 1$
    - balancePPp(t, r)
  - **else**
    - balancePPm(t, r)
  - $d := false$
- **else**
  - $t \rightarrow b := t \rightarrow b + 1$
  - $d := (t \rightarrow b > 0)$

balancePPp($\&t, r$: Node*)
$t \to right := r \to left$
$r \to left := t$
$r \to b := t \to b := 0$
$t := r$

balanceMMm($\&t, l$: Node*)
$t \to left := l \to right$
$l \to right := t$
$l \to b := t \to b := 0$
$t := l$



Szept 19.



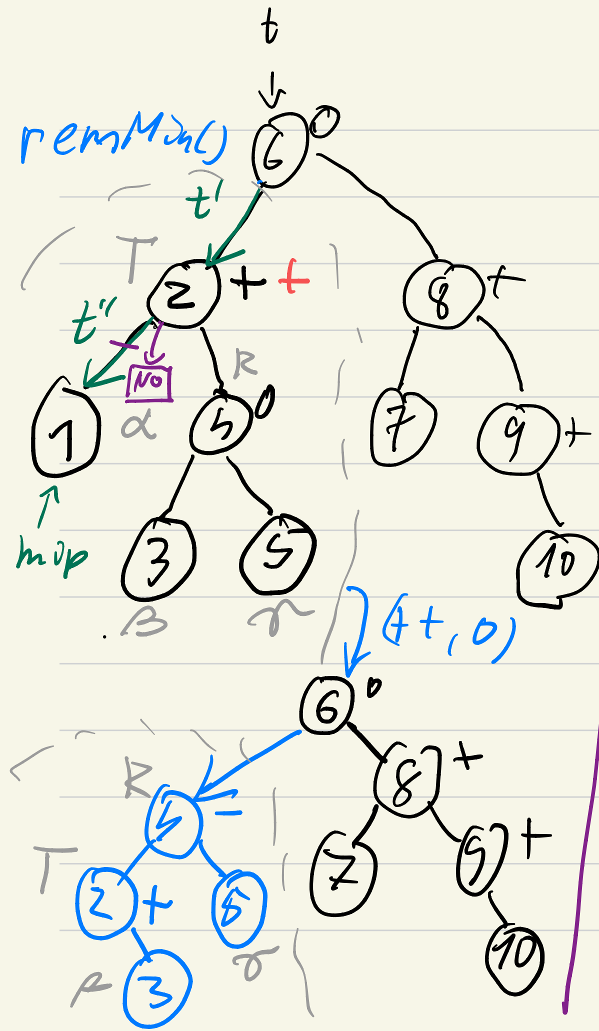
$\alpha < T < \beta < L < \gamma < R < \delta$

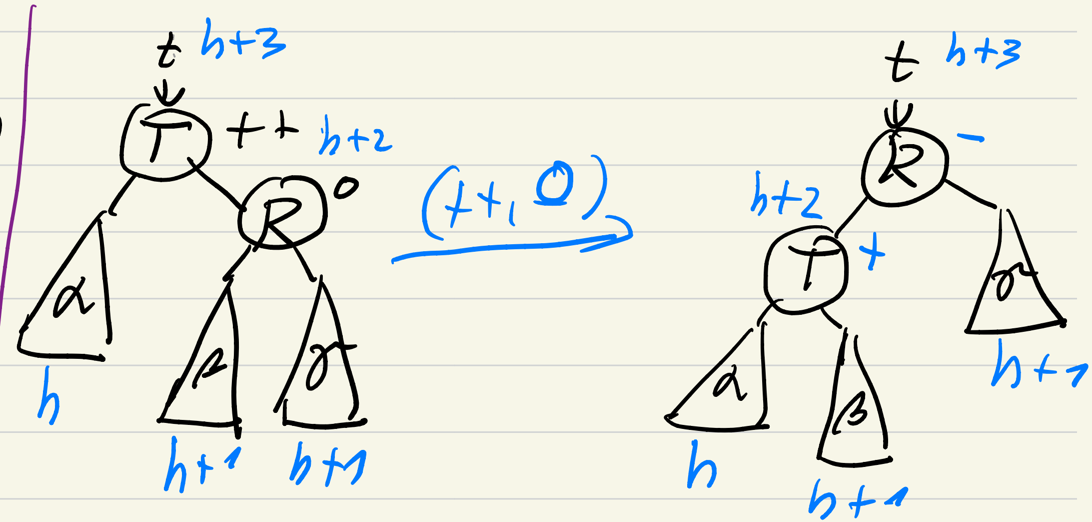
balancePPm(&t, r : Node*)

$l := r \to left$
$t \to right := l \to left$
$r \to left := l \to right$
$l \to left := t$
$l \to right := r$
$t \to b := -\lfloor (l \to b + 1)/2 \rfloor$
$r \to b := \lfloor (1 - l \to b)/2 \rfloor$
$l \to b := 0$
$t := l$

remMin(&t, &minp : Node*)

$t \to left = \emptyset$	
$minp := t$	remMin($t \to left, minp$)
$t := minp \to right$	
$minp \to right := \emptyset$	





$\alpha < T < \beta < R < \gamma$

AVLremMin(&t, &$minp$:Node* ; &d:$\mathbb{B}$)

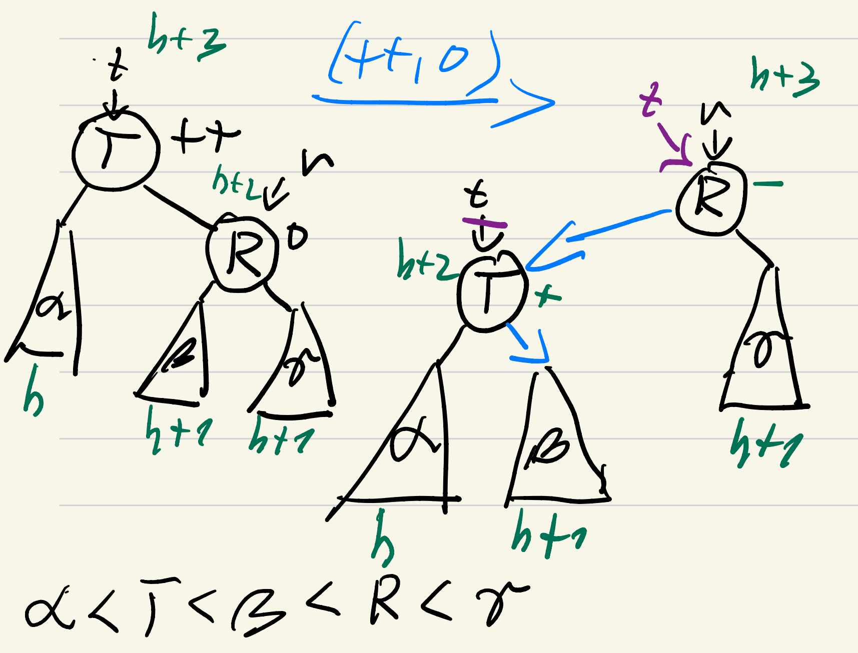
$t \to left = \emptyset$	
$minp := t$ $t := minp \to right$ $minp \to right := \emptyset$ $d := true$	AVLremMin$(t \to left, minp, d)$ d: leftSubTreeShrunk(t, d) | SKIP

leftSubTreeShrunk(&t:Node* ; &d:$\mathbb{B}$)

$t \to b = 1$	
// $t \to b := 2$ balance_PP(t, d)	$t \to b := t \to b + 1$ $d := (t \to b = 0)$

balance_PP(&t:Node* ; &d:$\mathbb{B}$)

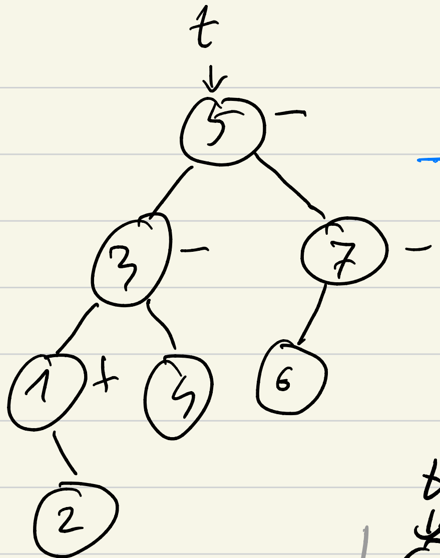
$r := t \to right$		
$r \to b = -1$	$r \to b = 0$	$r \to b = 1$
balancePPm(t, r)	balancePP0(t, r) $d := false$	balancePPp(t, r)



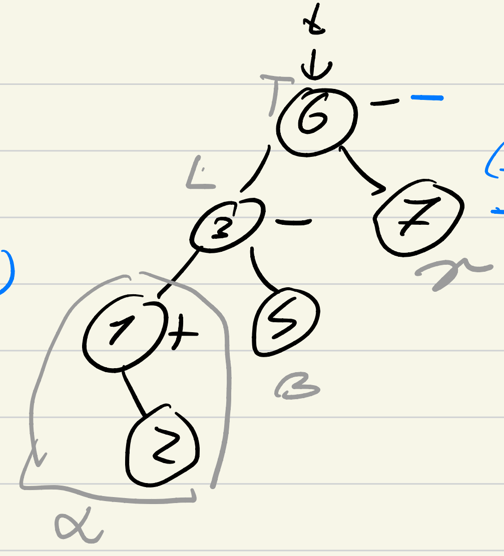
balancePP0(&t, r : Node*)

$t \to right := r \to left$
$r \to left := t$
$t \to b := 1$
$r \to b := -1$
$t := r$

del(t,5)



AVLremMin
(t->right, p, d)
1



(−, −, −)

$\alpha < L < \beta < T < \gamma$

(−, −, −)