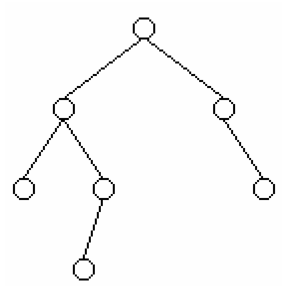
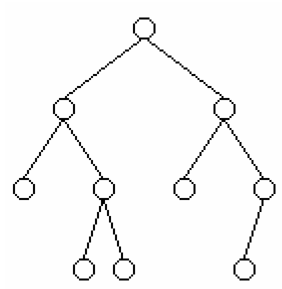
16. Az AVL-fa

(Adelszon-Velszkij és Landisz, 1962)

Definíció: t kiegyensúlyozott (AVL-tulajdonságú) $\Leftrightarrow$ t minden x csúcsára:
$|\,h(bal(x)) - h(jobb(x))\,| \leq 1$.

Pl.:





A majdnem teljes bináris fa AVL-tulajdonságú.

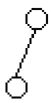
Az AVL-fára, mint speciális alakú keresőfára, változatlanul érvényesek a levezetett műveletek.
Minden művelet (beszúrás és törlés) után ellenőrizzük, és ha kell, helyreállítjuk az AVL-tulajdonságot.

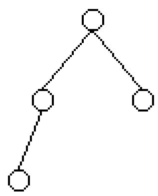
Belátjuk: 1. Az AVL-fa maximális magassága: $1{,}44 \log_2 n$.
2. Az AVL-tulajdonság helyreállítása $\Theta(1)$ $(illetve\, O(n \log n))$ művelettel!

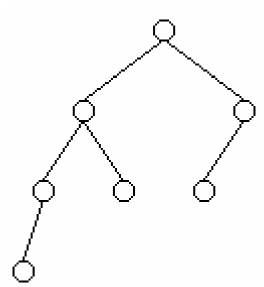
Legkevesebb pontszámú legmagasabb fa: Fibonacci-fa

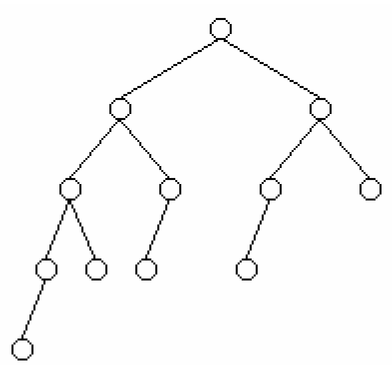
T_0 Ω 0

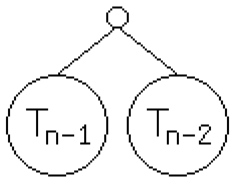
T_1  1

T_2  2

T_3  4

T_4  7

T_5  12

T_n 

Összefüggés az AVL-fa pontszáma és magassága között:

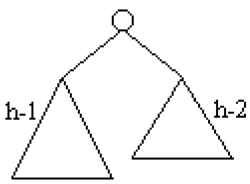
1) n adattal felépíthető fa minimális magassága? $\Rightarrow$ Majdnem teljes bináris fa
2) n adattal felépíthető fa maximális magassága?

Ugyanez máshogy: az adott h magasságú AVL-fák között mennyi a minimális pontszám?

											3								5
n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	…	19	20
n	0		1			2										4			

Jelölje $n(h)$ a h magasságú AVL-fák minimális pontszámát. Pl.: $n(4)=12$.
Rekurzív összefüggés:

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



$n(0)$

$n(1)=2$

Legyen g(h)=g(h-1)+g(h-2) - Fionacci-sorozat; $F_0=0, F_1=1, F_2=1$

g(0)=2
g(1)=3

A g(h) sorozat a „3-mal eltolt” Fibonacci-sorozat:

g(h)=F_{h+3}

Ismeretes, hogy $F_n=\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n-\left(\frac{1-\sqrt{5}}{2}\right)^n\right]$

$$n(h) = g(h) - 1 = \frac{1}{\sqrt{5}}\left[\left(\left(\frac{1+\sqrt{5}}{2}\right)^{h+3} - \left(\frac{1-\sqrt{5}}{2}\right)^{h+3}\right)\right] - 1 \geq \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{3}\left(\frac{1+\sqrt{5}}{2}\right)^{h} - 2 =$$

$$\frac{1}{\sqrt{5}}\frac{1+3\sqrt{5}+3\cdot 5+5\sqrt{5}}{8}\left(\frac{1+\sqrt{5}}{2}\right)^{h} - 2 = \frac{2+\sqrt{5}}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{h} - 2 =$$

$$= \left(\frac{1+\sqrt{5}}{2}\right)^{h} + \frac{2}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{h} - 2 \geq \left(\frac{1+\sqrt{5}}{2}\right)^{h} \qquad / \log_2$$

$$h \leq \frac{1}{\log_2\left(\frac{1+\sqrt{5}}{2}\right)} \cdot \log_2 n(h) \;\Rightarrow\; h \leq 1{,}44 \log_2 n$$

$\log_2 n(h)$: a h-hoz tartozó minimális pontszám

h: az n-hez tartozó h

Meggondolható, hogy eredményünk alapján a „fordított” kérdésre is érvényes a válasz.
Ez azért kedvező, mert láttuk, hogy a keresőfa műveletei

$$T(t) = \mathrm{O}(h(t))$$

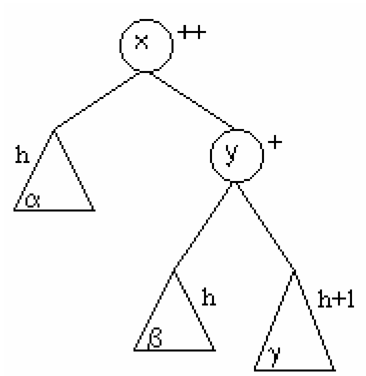
$$\downarrow \qquad \text{(mostani eredményeket figyelembe véve)}$$

$$T(n) = \mathrm{O}(\log n)$$

Kérdés: Az AVL-tulajdonság milyen ráfordítással tartható fenn?

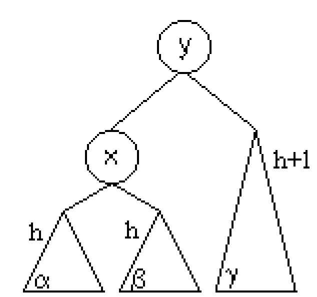
A hibás esetek megfoghatók 2 sémában (+ tükörképeik):

A) (++, +) szabály



Ez jelzi, hogy elromlott az AVL-tulajdonság

$\Rightarrow$

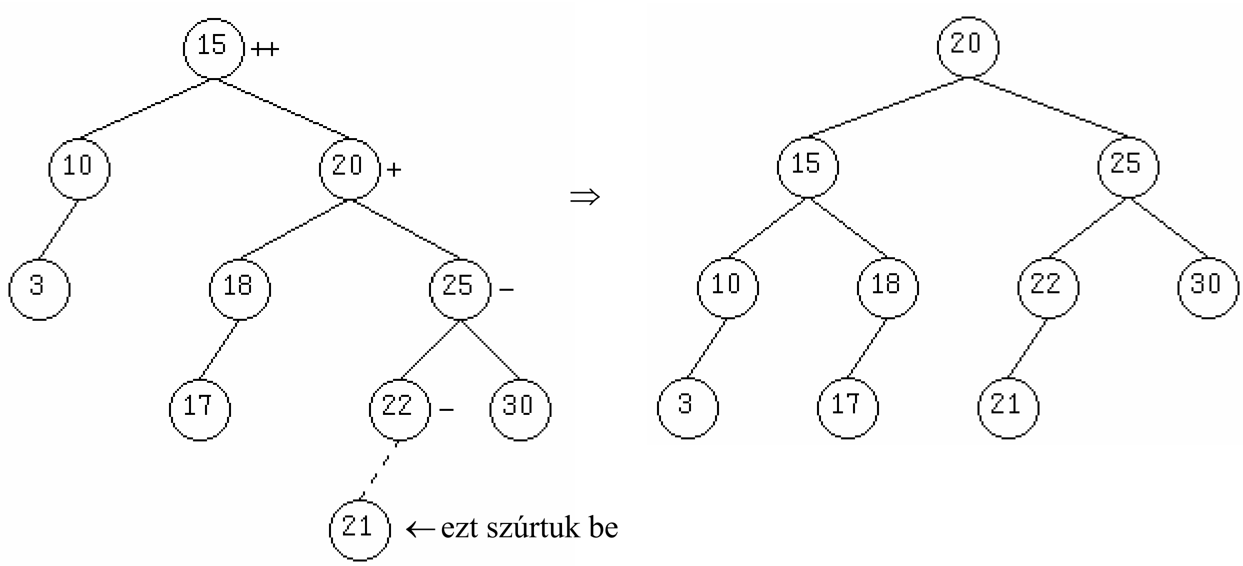


$\alpha < x < \beta < y < \gamma$

Ezt a transzformációt (és a többbit is) forgatásnak nevezik.

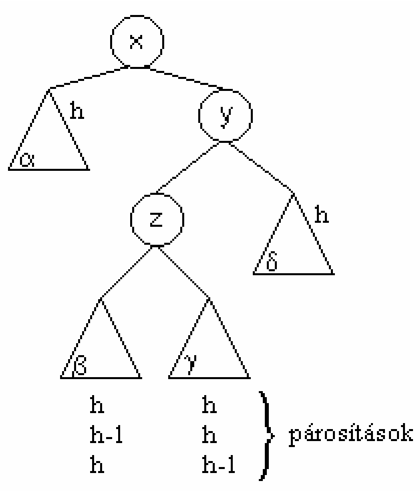
Ennek a tükörképe a (--, -) szabály.

Pl.:

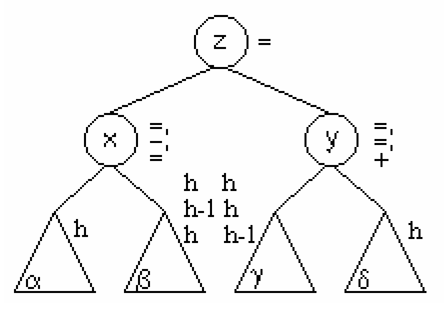


Ha a fában nem szerepelne a 17, akkor a 20^{++} romlana el, és csak ezt kellene helyreállítani.
Induljunk el a beszúrt csúcs szülőjétől a gyökér felé. Addig menjünk, amíg (a korrekciókat elvégezve), = vagy ++ nem jön, illetve amíg nem változott csúcshoz nem érünk (legfeljebb a gyökérig); a ++ esetben javítunk, ott = lesz, és tovább már nem kell nézni.

B) (++, -) szabály



⇒

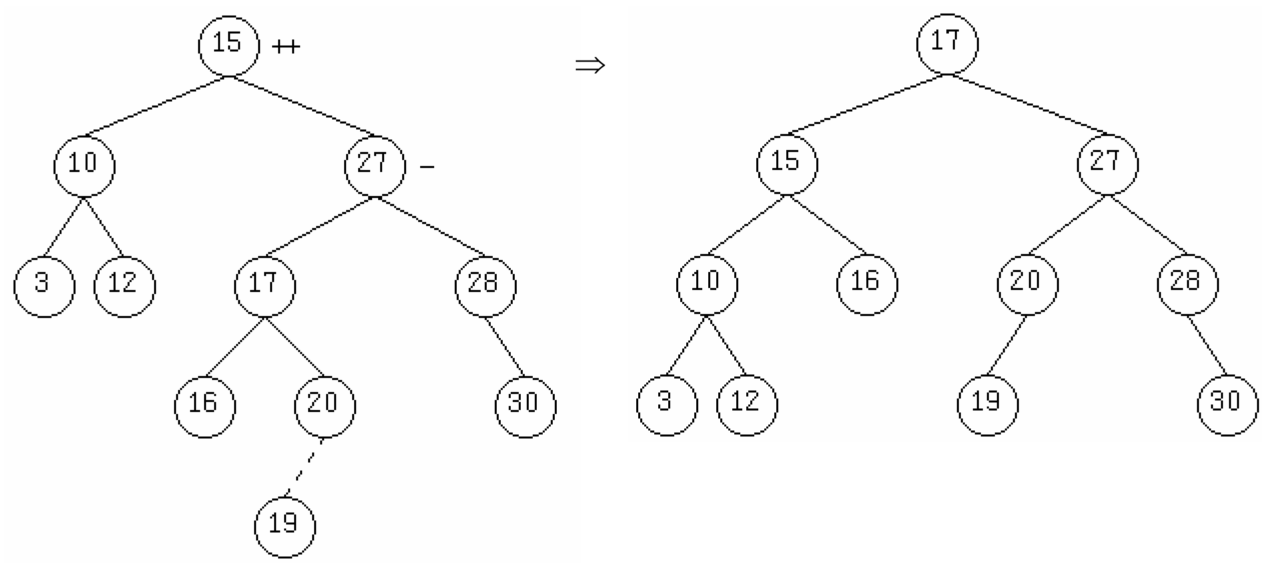


Forgatás

$\alpha < x < \beta < z < \gamma < y < \delta$

Ennek a tükörképe a (--, +) szabály.

Pl.:



Hogyan kell ábrázolni az indikátorokat?
Vagy 2 biten 3 információ vagy 2 byte-on a 2 magasságérték.

Törlésnél a fizikailag törölt csúcs szülőjénél kezdjük a korrekciót. (Ide kiegészítés tartozik)

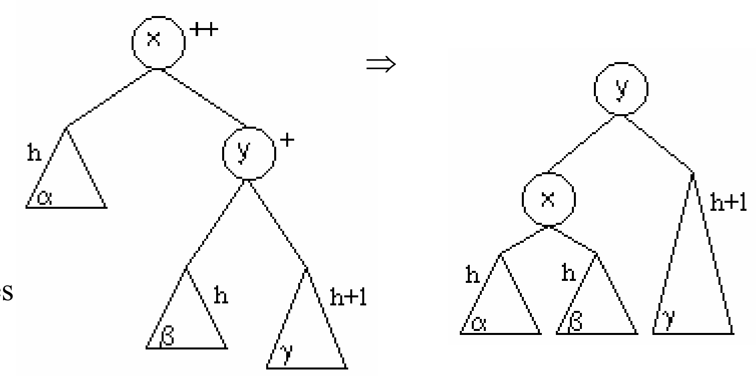
Az AVL-tulajdonság ellenőrzése egy útvonal bejárásával elvégezhető, ami $O(\log n)$ idejű. A forgatások pedig rendre 6 illetve 10 pointer állítását igénylik, ez $O(1)$ műveletigény. Végül tehát $O(\log n) + O(\log n) + O(1) = O(\log n)$ marad a műveletigény.

Kiegészítés

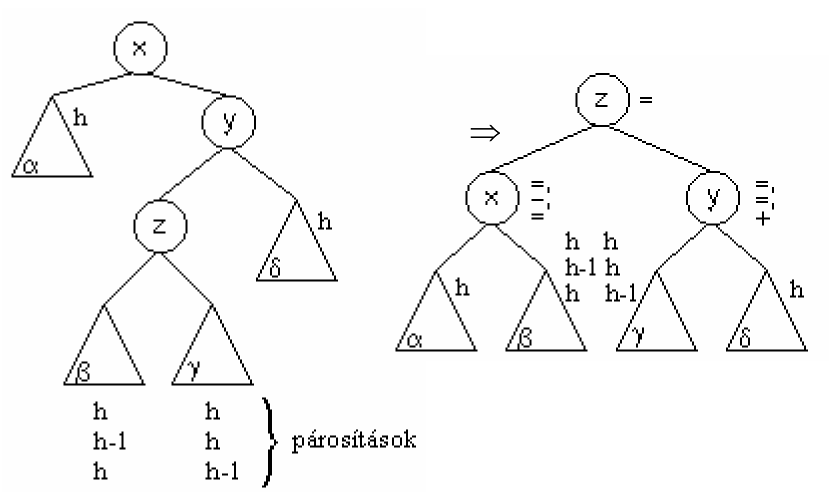
Megvizsgáljuk, hogyan lehet helyreállítani egy elromlott AVL-fát.

1. Beszúrás

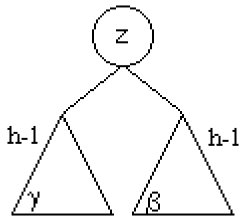
A. Ha a beszúrás hatására (++, +) típusú lett az AVL-fa, akkor az új levél a γ részfába került. A beszúrás előtt a fa magassága h+2 volt. A forgatás után ismét h+2 lett a magasság. Ezért feljebb, a fában (ha a létezik) változatlanul érvényes az AVL-tulajdonság; nem kell feljebb menni ellenőrizni és nem kell forgatni.



B. Ha a beszúrás után elromlott fa típusa (++, -) típusú, akkor az új levél a „z” alatti β-ba vagy γ-ba került.



Előzőleg a z csúcs alatt így nézett ki a fa: $|\beta|=|\gamma|=h-1$



A beszúrás után az egyik részfa magassága h lett, a másik maradt h-1.

Látható, hogy a beszúrás előtt az x gyökerű fa magassága h+2 volt, a forgatás után ismét h+2 lesz. Ezért feljebb, ha van befoglaló fa, ott már nem romolhat el az AVl-tulajdonság.

Összefoglalva: A beszúrás után az új levéltől felfelé haladva a gyökér felé újra számoljuk a csúcsok címkéit ezen az útvonalon. Ha egy x csúcs címkéje ++ vagy – lesz, akkor az x gyökerű (rész)fa forgatásával helyreállítható az AVL-tulajdonság. Beszúrás után tehát mindig elegendő egyetlen forgatás, ha elromlott a fa. Műveletigény $\Theta(1)$.

2. Törlés

Használjuk az előző ábrákat!

A. (++, +) A törlés α történt. Az α részfa magassága (h+1)volt és h lett. Az x gyökerű fa magassága (h+3)-ról (h+2)-re csökkent.

B. (++, -) A törlés itt is α -ban történt. Az α magassága h+1 volt, h lett. Az x gyökerű fa magassága (h+3)- ról (h+2)-re csökkent.

(Ebben az esetben lehet $|\beta|=|\gamma|$=h.)

Összefoglalva:

Mivel az x gyökerű fa magassága csökkent a forgatással, ezért feljebb is, ha van befoglaló fa, elromolhatott az AVL-tulajdonság. Mindkét esetben.

Törlés után a törölt elem szülőjétől kezdve elindulunk felfelé és újra számoljuk a csúcsok címkéjét, ezen az útvonalon. Ha egy x csúcs címkéje ++ vagy – lesz, akkor az x gyökerű (részfa) forgatásával helyreállíthatjuk annak AVL-tulajdonságát. Ha x nem a gyökér, akkor feljebb kell lépni és folytatni kell az ellenőrzést. Szélsőséges esetben az adott útvonal minden pontjában forgatni kell. Ez történik, ha Fibonacci-fa legkisebb magasságú „jobb szélső" elemét töröljük. Műveletigény: O($n \log n$)

.