Arbori AVL

Universitatea "Transilvania" din Brașov

23 martie 2021

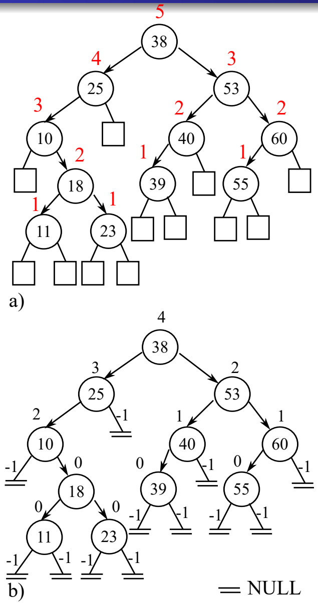
Arbori AVL - Factor de balansare

Arbori AVL:

- arbori binari de căutare aproape balansați.
- au primit denumirea după autori, matematicienii ruși G.M. Adelson-Velsky și E.M. Landis.
- fiecare nod x dispune de o proprietate *factor de balansare* dată prin

$$fb(x) = h(x.dr) - h(x.st)$$

Arbori AVL - Înălțime - Factor de balansare

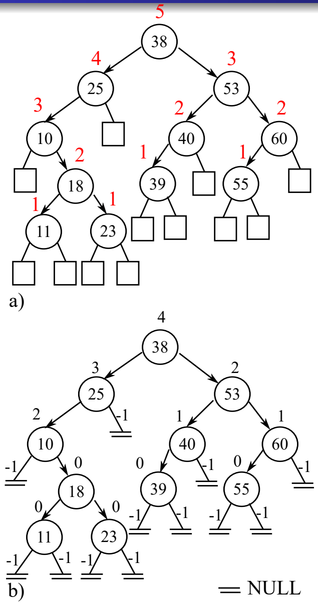


Înălțimea unui nod (subarbore) = lungimea celui mai lung drum de la nod la o frunză ⇒ înălțimea unei frunze este 0.

Pentru calculul corect al factorilor de balansare într-u AVL există două variante:

- se consideră că frunzele având copii noduri nule ⇒ înălțimea unei frunze va fi 1, iar a unui ”nod nul” va fi 0. (fig a.)

Arbori AVL - Înălțime - Factor de balansare

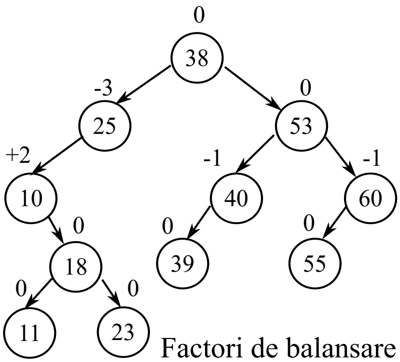


Înălțimea unui nod (subarbore) = lungimea celui mai lung drum de la nod la o frunză $\Rightarrow$ înălțimea unei frunze este 0.

Pentru calculul corect al factorilor de balansare într-u AVL există două variante:

- se consideră că frunzele având copii noduri nule $\Rightarrow$ înălțimea unei frunze va fi 1, iar a unui ”nod nul” va fi 0. (fig a.)
- se consideră înălțmea unei frunze =0, iar pentru ”NULL” se consideră ”înălțimea” -1 (fig b.)

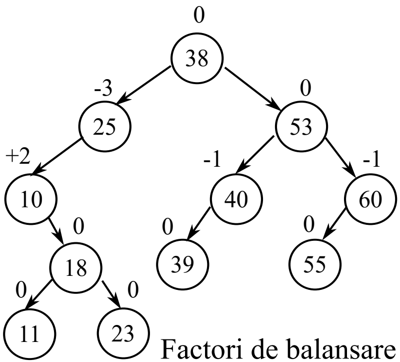
Arbori AVL - Factor de balansare



Nodurile 10 și 25 NU sunt balansate.

Nod balansat: Un nod x se numște balansat, dacă $fb(x) \in \{-1, 0, 1\}$.

Arbori AVL - Factor de balansare



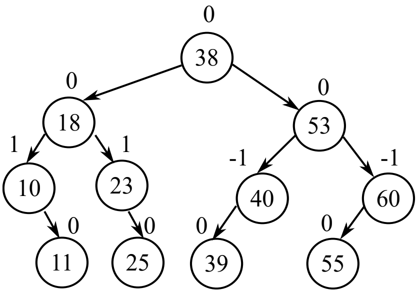
Factori de balansare

Nodurile 10 și 25 NU sunt balansate.
Nu este arbore AVL

Nod balansat: Un nod x se numște balansat, dacă $fb(x) \in \{-1, 0, 1\}$.

Arbore AVL: Un arbore binar de căutare se numește *arbore AVL*, dacă fiecare nod al său este balansat.

Arbori AVL - Factor de balansare



Arbore AVL

Nod balansat: Un nod x se numște balansat, dacă $fb(x) \in \{-1, 0, 1\}$.

Arbore AVL: Un arbore binar de căutare se numește *arbore AVL*, dacă fiecare nod al său este balansat.

Înălțimea unui arbore AVL

Demonstrăm faptul că: $h_{AVL} < 2\log_2 n$, $n =$ nr de noduri din AVL.

Considerăm un AVL cu n noduri și înălțimea h.

Înălțimea unui arbore AVL

Demonstrăm faptul că: $h_{AVL} < 2\log_2 n$, $n =$ nr de noduri din AVL.

Considerăm un AVL cu n noduri și înălțimea h.
Notăm $N(h)$ numărul minim de noduri al unui AVL de înălțime $h \Rightarrow n > N(h)$

Înălțimea unui arbore AVL

Demonstrăm faptul că: $h_{AVL} < 2\log_2 n$, $n =$ nr de noduri din AVL.

Considerăm un AVL cu n noduri și înălțimea h.
Notăm $N(h)$ numărul minim de noduri al unui AVL de înălțime $h \Rightarrow n > N(h)$
În plus factorul de balansare al rădăcinii r unui astfel de AVL este -1 sau 1.

Înălțimea unui arbore AVL

Demonstrăm faptul că: $h_{AVL} < 2\log_2 n$, $n =$ nr de noduri din AVL.

Considerăm un AVL cu n noduri și înălțimea h.
Notăm $N(h)$ numărul minim de noduri al unui AVL de înălțime $h \Rightarrow n > N(h)$
În plus factorul de balansare al rădăcinii r unui astfel de AVL este -1 sau 1.
$\Rightarrow$ h(r.st) $=$ h(r.dr) $+1$ sau invers și ambii subarbori au număr minim de noduri

Înălțimea unui arbore AVL

Demonstrăm faptul că: $h_{AVL} < 2\log_2 n$, $n =$ nr de noduri din AVL.

Considerăm un AVL cu n noduri și înălțimea h.
Notăm $N(h)$ numărul minim de noduri al unui AVL de înălțime $h \Rightarrow n > N(h)$
În plus factorul de balansare al rădăcinii r unui astfel de AVL este -1 sau 1.
$\Rightarrow$ h(r.st) $=$ h(r.dr) $+1$ sau invers și ambii subarbori au număr minim de noduri

$$\Rightarrow N(h) = N(h-1) + N(h-2) + 1$$

Înălțimea unui arbore AVL

Demonstrăm faptul că: $h_{AVL} < 2\log_2 n$, $n =$ nr de noduri din AVL.

Considerăm un AVL cu n noduri și înălțimea h.
Notăm $N(h)$ numărul minim de noduri al unui AVL de înălțime $h \Rightarrow n > N(h)$
În plus factorul de balansare al rădăcinii r unui astfel de AVL este -1 sau 1.
$\Rightarrow$ h(r.st) $=$ h(r.dr) $+1$ sau invers și ambii subarbori au număr minim de noduri

$$\Rightarrow N(h) = N(h-1) + N(h-2) + 1$$

Cazurile de bază sunt:

$$N(0) = 1, N(1) = 1$$

și $N(h-1) < N(h-2) \Rightarrow$

Înălțimea unui arbore AVL

Demonstrăm faptul că: $h_{AVL} < 2\log_2 n$, $n =$ nr de noduri din AVL.

Considerăm un AVL cu n noduri și înălțimea h.
Notăm $N(h)$ numărul minim de noduri al unui AVL de înălțime $h \Rightarrow n > N(h)$
În plus factorul de balansare al rădăcinii r unui astfel de AVL este -1 sau 1.
$\Rightarrow$ h(r.st) = h(r.dr) +1 sau invers și ambii subarbori au număr minim de noduri

$$\Rightarrow N(h) = N(h-1) + N(h-2) + 1$$

Cazurile de bază sunt:

$$N(0) = 1, N(1) = 1$$

și $N(h-1) < N(h-2) \Rightarrow$

$$\Rightarrow N(h) > 2N(h-2) > 4N(h-4) > 8N(h-6) > \ldots > 2^k N(h-2k) > \ldots > 2^{h/2}$$

Înălțimea unui arbore AVL

Demonstrăm faptul că: $h_{AVL} < 2\log_2 n$, n = nr de noduri din AVL.

Considerăm un AVL cu n noduri și înălțimea h.
Notăm $N(h)$ numărul minim de noduri al unui AVL de înălțime $h \Rightarrow n > N(h)$
În plus factorul de balansare al rădăcinii r unui astfel de AVL este -1 sau 1.
$\Rightarrow$ h(r.st) = h(r.dr) +1 sau invers și ambii subarbori au număr minim de noduri

$$\Rightarrow N(h) = N(h-1) + N(h-2) + 1$$

Cazurile de bază sunt:

$$N(0) = 1, N(1) = 1$$

și $N(h-1) < N(h-2) \Rightarrow$

$$\Rightarrow N(h) > 2N(h-2) > 4N(h-4) > 8N(h-6) > \ldots > 2^k N(h-2k) > \ldots > 2^{h/2}$$

$$\Rightarrow n \geq N(h) > 2^{h/2}$$

Înălțimea unui arbore AVL

Demonstrăm faptul că: $h_{AVL} < 2\log_2 n$, $n =$ nr de noduri din AVL.

Considerăm un AVL cu n noduri și înălțimea h.
Notăm $N(h)$ numărul minim de noduri al unui AVL de înălțime $h \Rightarrow n > N(h)$
În plus factorul de balansare al rădăcinii r unui astfel de AVL este -1 sau 1.
$\Rightarrow$ h(r.st) $=$ h(r.dr) $+1$ sau invers și ambii subarbori au număr minim de noduri

$$\Rightarrow N(h) = N(h-1) + N(h-2) + 1$$

Cazurile de bază sunt:

$$N(0) = 1, N(1) = 1$$

și $N(h-1) < N(h-2) \Rightarrow$

$$\Rightarrow N(h) > 2N(h-2) > 4N(h-4) > 8N(h-6) > \ldots > 2^k N(h-2k) > \ldots > 2^{h/2}$$

$$\Rightarrow n \geq N(h) > 2^{h/2}$$

$$\Rightarrow h < 2log_2(n)$$

Arbore AVL - complexitatea operațiilor

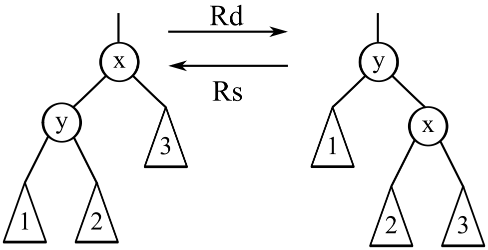
- Într-un AVL cu n noduri și înălțimea h s-a demonstrat:

$$h < 2log_2(n)$$

- Complexitatea operațiilor de bază (căutare, inserție, ștergere) depinde de înălțimea arborelui

Deci complexitatea operațiilor de bază este $O(\log_2 n)$.

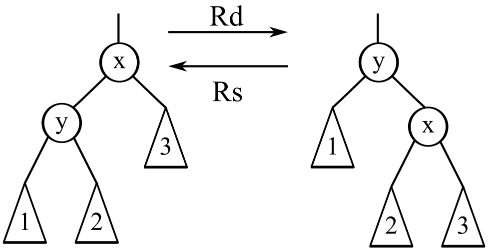
Rotația



Rotația într-un arbore binar de căutare: operație locală care

- schimbă structura locală de pointeri
- păstrează proprietățile de arbore de căutare.

Rotația



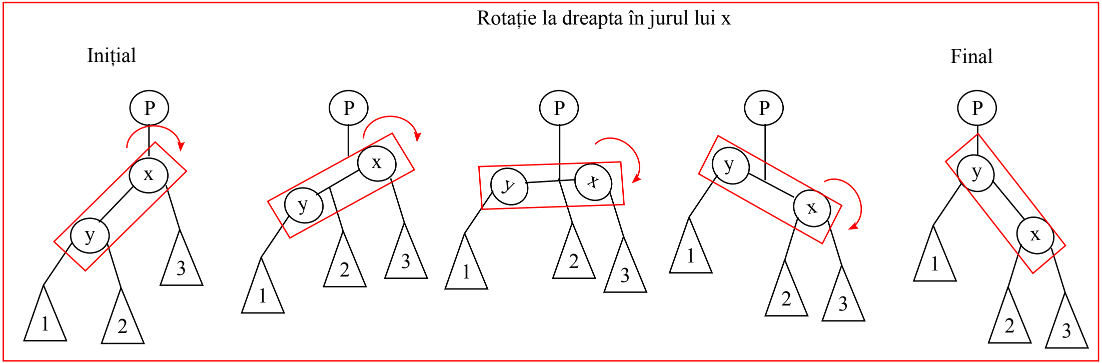
Rotația într-un arbore binar de căutare: operație locală care

- schimbă structura locală de pointeri
- păstrează proprietățile de arbore de căutare.

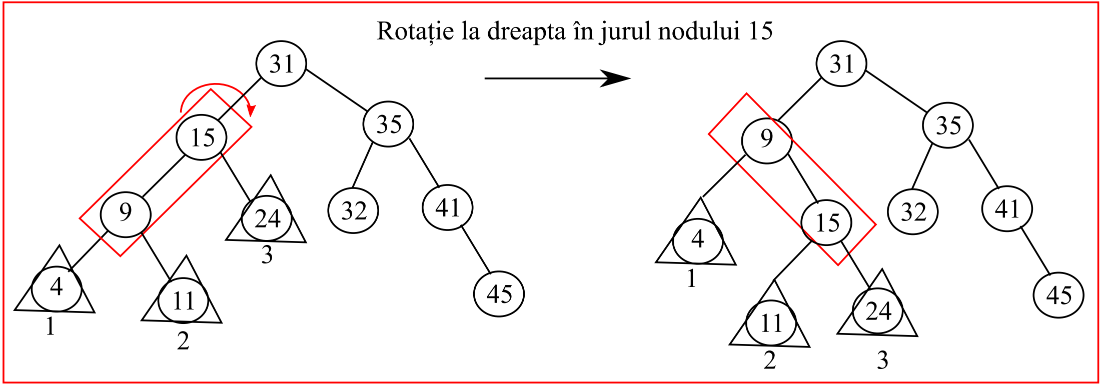
Observații:

- pentru rotație la stânga în jurul lui x trebuie ca $x.dr \neq$ NULL
- pentru rotație la dreapta în jurul lui x trebuie ca $x.st \neq$ NULL

Rotația



Rotația - Exemplu



Inserția

Inserearea unui nod nou x

- se realizează ca la arborii binari de căutare

Inserția

Inserearea unui nod nou x

- se realizează ca la arborii binari de căutare
- nodul inserat va fi frunză și are factorul de balansare 0

Inserția

Inserearea unui nod nou x

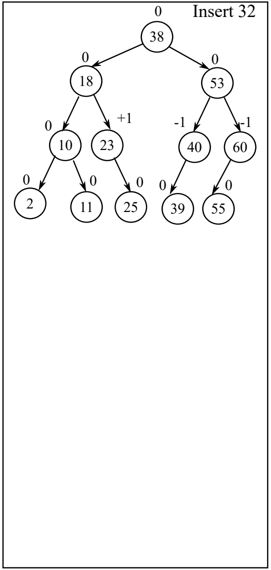
- se realizează ca la arborii binari de căutare
- nodul inserat va fi frunză și are factorul de balansare 0
- se recalculează factorii de balansare în manieră *bottom-up*, pornind de la x înspre rădăcină

Inserția

Inserearea unui nod nou x

- se realizează ca la arborii binari de căutare
- nodul inserat va fi frunză și are factorul de balansare 0
- se recalculează factorii de balansare în manieră *bottom-up*, pornind de la x înspre rădăcină
- dacă factorul nou al unui nod devine $+2$ sau -2, nodul trebuie rebalansat prin rotații

Inserția

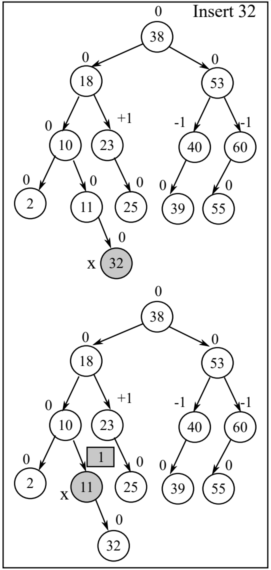


Recalcularea factorilor

Se inserează nodul x cu factorul 0.
Se pornește recalcularea factorilor de la $x = x.p$ în sus.
Notăm cu x nodul curent și se recalculează fb(x).

- dacă factorul de balansare al lui x devine $+1$ sau -1 atunci se urcă în arbore $(x = x.p)$ și se reia algoritmul.

Inserția

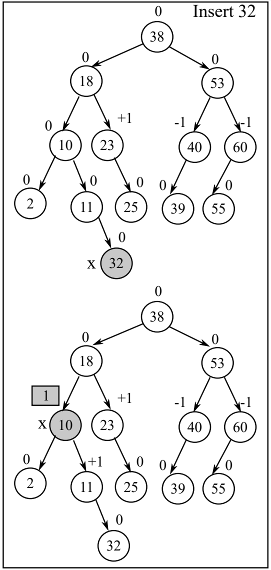


Recalcularea factorilor

Se inserează nodul x cu factorul 0.
Se pornește recalcularea factorilor de la $x = x.p$ în sus.
Notăm cu x nodul curent și se recalculează fb(x).

- dacă factorul de balansare al lui x devine $+1$ sau -1 atunci se urcă în arbore ($x = x.p$) și se reia algoritmul.

Inserția

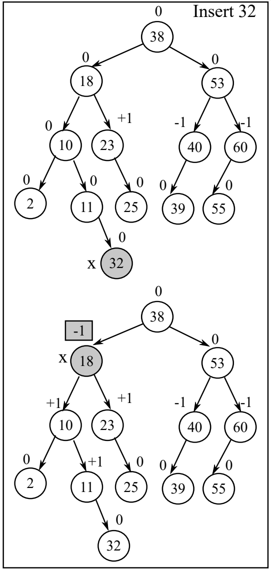


Recalcularea factorilor

Se inserează nodul x cu factorul 0.
Se pornește recalcularea factorilor de la $x = x.p$ în sus.
Notăm cu x nodul curent și se recalculează fb(x).

- dacă factorul de balansare al lui x devine $+1$ sau -1 atunci se urcă în arbore $(x = x.p)$ și se reia algoritmul.

Inserția

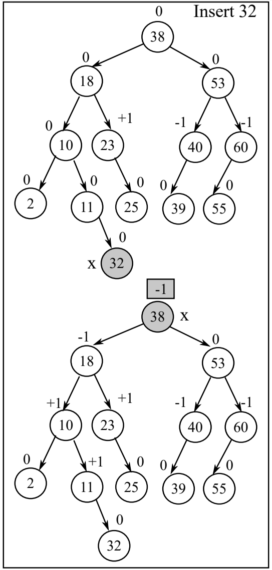


Recalcularea factorilor

Se inserează nodul x cu factorul 0.
Se pornește recalcularea factorilor de la $x = x.p$ în sus.
Notăm cu x nodul curent și se recalculează fb(x).

- dacă factorul de balansare al lui x devine $+1$ sau -1 atunci se urcă în arbore ($x = x.p$) și se reia algoritmul.

Inserția



Recalcularea factorilor

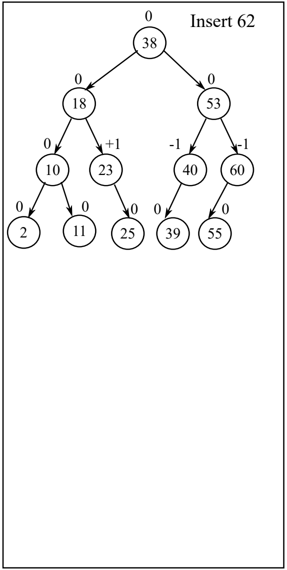
Se inserează nodul x cu factorul 0.

Se pornește recalcularea factorilor de la $x = x.p$ în sus.

Notăm cu x nodul curent și se recalculează fb(x).

- dacă factorul de balansare al lui x devine $+1$ sau -1 atunci se urcă în arbore ($x = x.p$) și se reia algoritmul.

Inserția

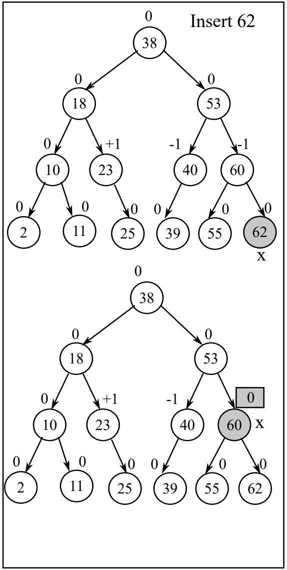


Recalcularea factorilor

Se inserează nodul x cu factorul 0.
Se pornește recalcularea factorilor de la $x = x.p$ în sus.
Notăm cu x nodul curent și se recalculează fb(x).

- dacă factorul de balansare al lui x devine $+1$ sau -1 atunci se urcă în arbore ($x = x.p$) și se reia algoritmul.
- dacă factorul de balansare al lui x devine 0 atunci STOP

Inserția

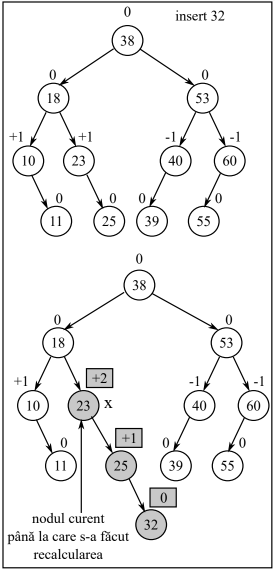


Recalcularea factorilor

Se inserează nodul x cu factorul 0.
Se pornește recalcularea factorilor de la $x = x.p$ în sus.
Notăm cu x nodul curent și se recalculează fb(x).

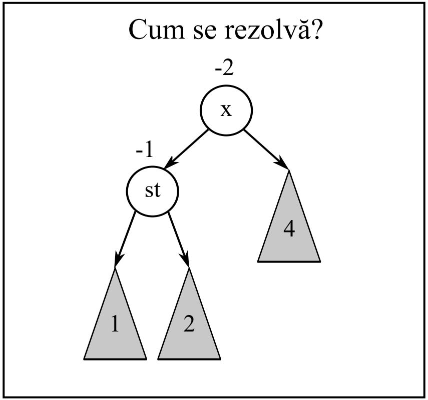
- dacă factorul de balansare al lui x devine $+1$ sau -1 atunci se urcă în arbore ($x = x.p$) și se reia algoritmul.
- dacă factorul de balansare al lui x devine 0 atunci STOP

Inserția



Dacă factorul de balansare nou al lui x este -2 sau $+2$, atunci s-a produs o debalansare a subarborelui de rădăcină x și sunt necesare proceduri de rebalansare.

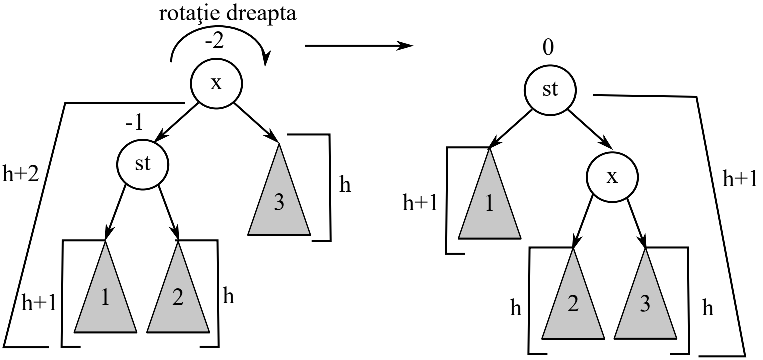
Inserția - Rebalansarea arborelui



Rebalansarea aborelui.: nodul curent este x.
Cazul 1:

Dacă $fb(x) = -2$ și $fb(x.st) = -1$ cum rebalansăm arborele?

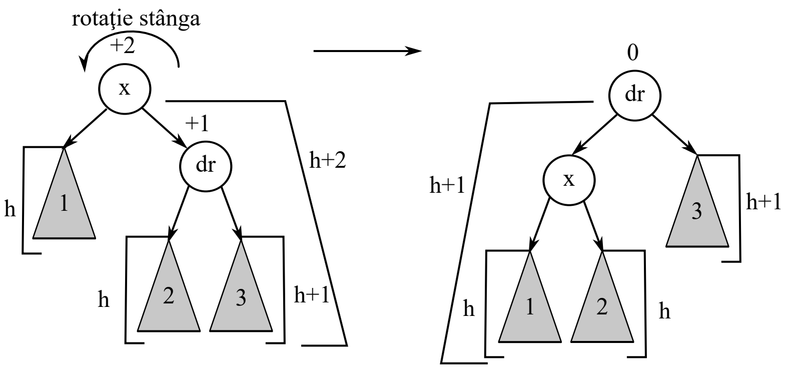
Inserția - Rebalansarea arborelui



Rebalansarea aborelui.: nodul curent este x.
Cazul 1:

a. $fb(x)$ este -2 și $fb(x.st) = -1 \Rightarrow$ rotație la dreapta în jurul lui x

Inserția - Rebalansarea arborelui



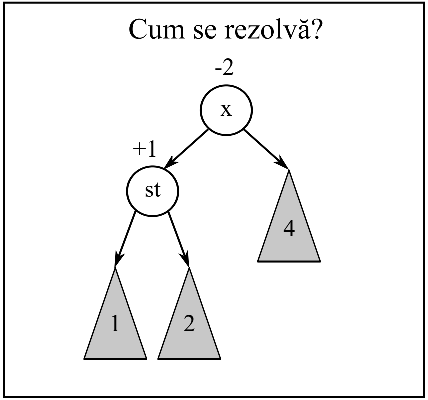
Rebalansarea aborelui.: nodul curent este x.

Cazul 1:

a. $fb(x)$ este -2 și $fb(x.st) = -1 \Rightarrow$ rotație la dreapta în jurul lui x

b. $fb(x)$ este 2 și $fb(x.dr) = 1 \Rightarrow$ rotație la stânga în jurul lui x

Inserția - Rebalansarea arborelui

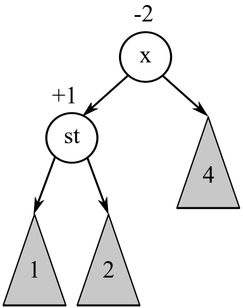


Rebalansarea aborelui.: nodul curent este x.
Cazul 2:

Dacă $fb(x) = -2$ și $fb(x.st) = 1$ cum rebalansăm arborele?

Inserția - Rebalansarea arborelui

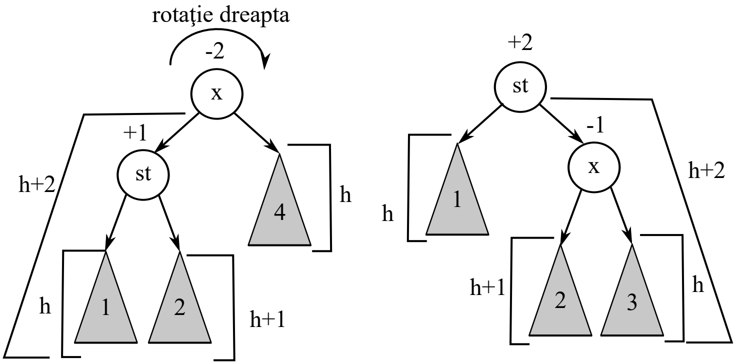
Cum se rezolvă?



Rebalansarea aborelui.: nodul curent este x.
Cazul 2:

Prima idee - rotație la dreapta în jurul lui x.

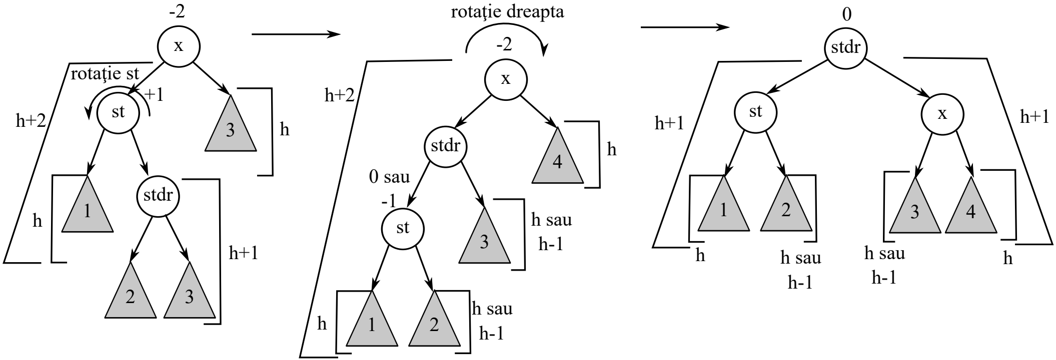
Inserția - Rebalansarea arborelui



Rebalansarea aborelui.: nodul curent este x.
Cazul 2:

Prima idee - rotație la dreapta în jurul lui x. $\Rightarrow$ se debalansează pe partea cealaltă $\Rightarrow$ nu e suficient!

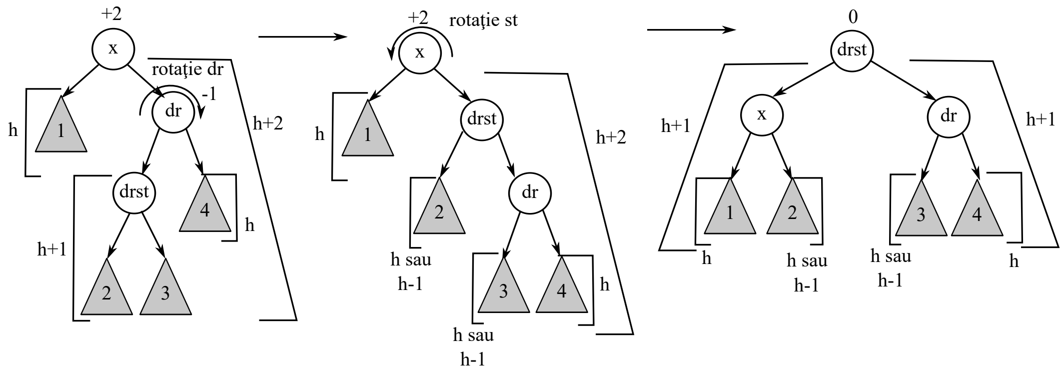
Inserţia - Rebalansarea arborelui



Rebalansarea aborelui.: nodul curent este x.
Cazul 2:

[a.] $fb(x)$ este -2 şi $fb(x.st) = 1 \Rightarrow$ rotaţie la stânga în jurul lui $x.st$, apoi rotaţie la dreapta în jurul lui x

Inserția - Rebalansarea arborelui



Rebalansarea aborelui.: nodul curent este x.
Cazul 2:

[b.] $fb(x)$ este 2 și $fb(x.dr) = -1 \Rightarrow$ rotație la dreapta în jurul lui $x.dr$, apoi rotație la stânga în jurul lui x

Ștergerea

Ștergerea nodului z: - etapa inițială = ștergerea lui z ca la un arbore binar de căutare obișnuit

a. Dacă z are un singur fiu (sau niciunul) - se înlocuiește z cu fiul respectiv (sau cu NULL)

b. Dacă z are doi fii - se determină $y = \text{succesor}(z)$, se înlocuiește z cu y și y cu $y.dr$.

Apoi se rebalansează arborele de jos în sus:

Ștergerea

Ștergerea nodului z: - etapa inițială = ștergerea lui z ca la un arbore binar de căutare obișnuit

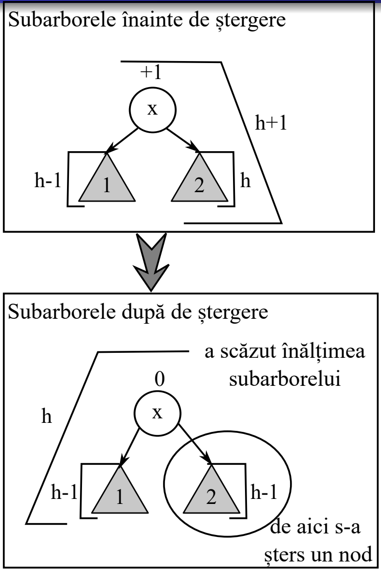
a. Dacă z are un singur fiu (sau niciunul) - se înlocuiește z cu fiul respectiv (sau cu NULL)

b. Dacă z are doi fii - se determină y = succesor(z), se înlocuiește z cu y și y cu $y.dr$.

Apoi se rebalansează arborele de jos în sus:

- În cazul (a.) rebalansarea începe de la părintele nodului șters z
- În cazul (b.) rebalansarea începe de la părintele succesorului y

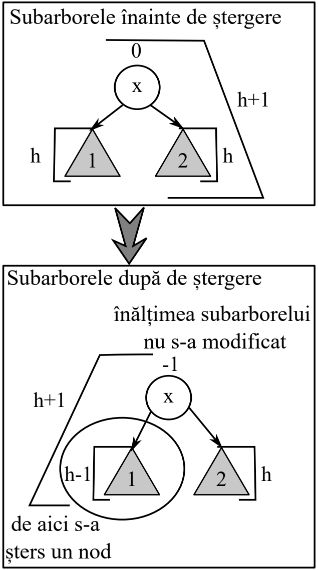
Ștergerea - Rebalansare



Notăm cu x factorul de balansare curent.

- dacă după recalculare $fb(x) = 0 \Rightarrow$ a scăzut înălțimea subarborelui de rădăcină $x \Rightarrow$ se continuă recalcularea factorilor și rebalansarea de la $x = x.p$.

Ștergerea - Rebalansare

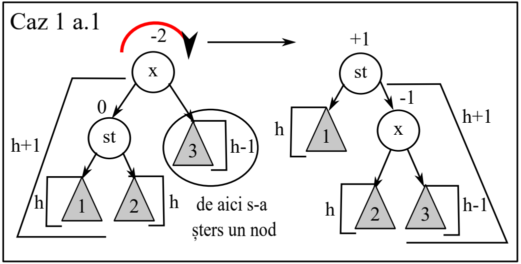


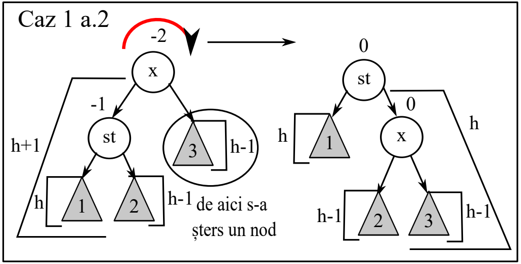
Notăm cu x factorul de balansare curent.

- dacă după recalculare $fb(x) = 0 \Rightarrow$ a scăzut înălțimea subarborelui de rădăcină $x \Rightarrow$ se continuă recalcularea factorilor și rebalansarea de la $x = x.p$.

- dacă după recalculare $fb(x) = -1$ sau $fb(x) = 1 \Rightarrow$ înălțimea subarborelui de rădăcină x nu se modifică $\Rightarrow$ STOP

Ștergerea - Rebalansare

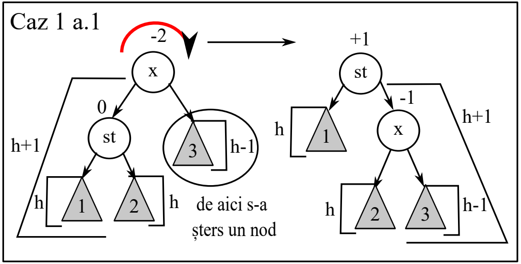


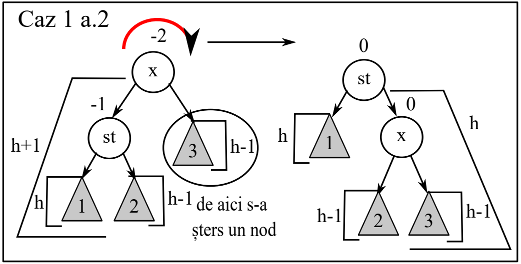


Cazul 1: Notăm cu x factorul de balansare curent.

- dacă după recalculare $fb(x) = -2$ și pentru $st = x.st$ avem $fb(st) = 0 \Rightarrow$ rotație la dreapta după x și STOP.

Ștergerea - Rebalansare



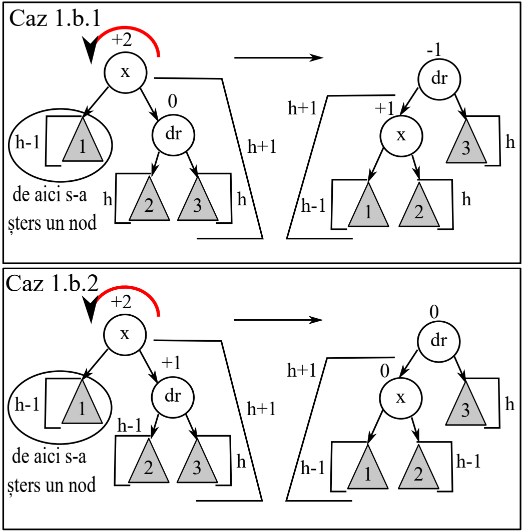


Cazul 1: Notăm cu x factorul de balansare curent.

- dacă după recalculare $fb(x) = -2$ și pentru $st = x.st$ avem $fb(st) = 0 \Rightarrow$ rotație la dreapta după x și STOP.

- dacă după recalculare $fb(x) = -2$ și pentru $st = x.st$ avem $fb(st) = -1 \Rightarrow$ rotație la dreapta după x și continui de la $x = x.p$

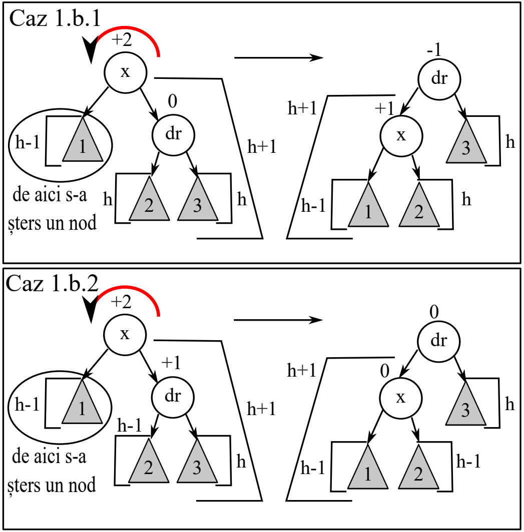
Ștergerea - Rebalansare



Cazul 1: Notăm cu x factorul de balansare curent.

- dacă după recalculare $fb(x) = +2$ și pentru $dr = x.dr$ avem $fb(dr) = 0 \Rightarrow$ rotație la stânga după x și STOP.

Ștergerea - Rebalansare

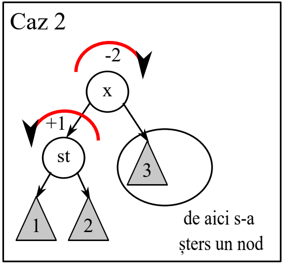


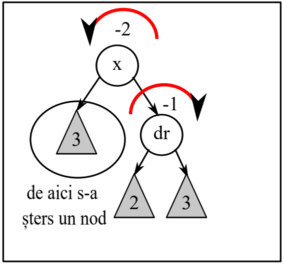
Cazul 1: Notăm cu x factorul de balansare curent.

- dacă după recalculare $fb(x) = +2$ și pentru $dr = x.dr$ avem $fb(dr) = 0 \Rightarrow$ rotație la stânga după x și STOP.

- dacă după recalculare $fb(x) = +2$ și pentru $dr = x.dr$ avem $fb(dr) = +1 \Rightarrow$ rotație la stânga după x și continui de la $x = x.p$

Ștergerea - Rebalansare

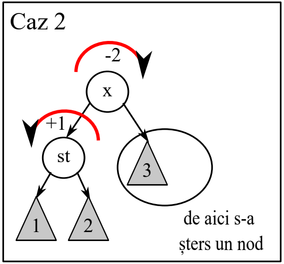


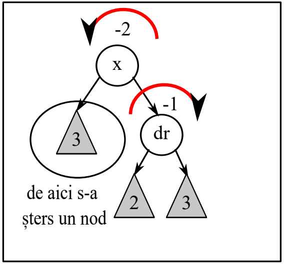


Cazul 2: Notăm cu x factorul de balansare curent.

- dacă după recalculare $fb(x) = -2$ și pentru $st = x.st$ avem $fb(st) = 1 \Rightarrow$ (1) rotație la stânga după st, apoi (2) rotație la dreapta după x și se continuă de la $x = x.p$.

Ștergerea - Rebalansare





Cazul 2: Notăm cu x factorul de balansare curent.

- dacă după recalculare $fb(x) = -2$ și pentru $st = x.st$ avem $fb(st) = 1 \Rightarrow$ (1) rotație la stânga după st, apoi (2) rotație la dreapta după x și se continuă de la $x = x.p$.

- dacă după recalculare $fb(x) = +2$ și pentru $dr = x.drt$ avem $fb(dr) = -1 \Rightarrow$ (1) rotație la dreapta după dr, apoi rotație la stânga după x și continui de la $x = x.p$