Algoritmuselmélet

Piros-fekete fák

Katona Gyula Y. / Vizer Máté

Számítástudományi és Információelméleti Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Alsó korlát bináris keresőfák magasságára

Múltkor már láttuk, hogy az n elemet tartalmazó bináris kereső fa magassága legrosszabb esetben $n-1$.

Azt is láttuk, hogy egy ℓ szintű bináris keresőfában legfeljebb $2^\ell - 1$ elem lehet (ilyenkor teljes a fa). $\implies n \leq 2^\ell - 1 \implies \lceil \log_2(n+1) \rceil \leq \ell$

Állítás

Egy n elemet tartalmazó bináris keresőfa magassága legalább $\lceil \log_2(n+1) \rceil - 1$.

Szeretnénk valahogy garantálni, hogy ennél ne legyen sokkal magasabb a fa. $\implies$ kiegyensúlyozott bináris keresőfák

Piros-fekete fák

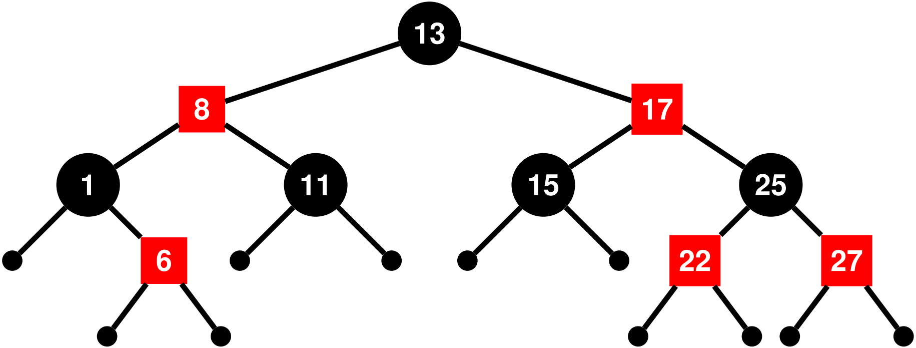
Olyan bináris keresőfa, melynek mélysége nem lehet nagy.
BESZÚR, TÖRÖL, KERES, MIN, MAX hatékonyak.

Definíció

*A **piros-fekete fa** egy bináris keresőfa, melyre teljesülnek a következők:*

1. *Minden nem levél csúcsnak 2 gyereke van.*
2. *Elemeket belső csúcsokban tárolunk.*
3. *Teljesül a keresőfa tulajdonság.*
4. *A fa minden csúcsa piros vagy fekete.*
5. *A gyökér fekete.*
6. *A levelek feketék.*
7. *Minden piros csúcs mindkét gyereke fekete.*
8. *Minden v csúcsra igaz, hogy az összes v-ből levélbe vezető úton ugyanannyi fekete csúcs van.*

Példa



Megj.: A szokásos bináris fát kiegészítjük üres levelekkel.

Piros-fekete fák

Jelölések

- F_v: v gyökerű részfa
- $m(v)$: v magassága, a leghosszabb v-ből levélbe vezető út éleinek száma
- $fm(v)$: v fekete-magassága, a v-ből levélbe vezető bármelyik úton a fekete csúcsok száma, v-t nem számolva.
  (Ez minden úton egyforma a 8. tulajdonság miatt.)

Tulajdonságok

Állítás

Egy piros-fekete fa minden v csúcsára teljesül

$$\frac{m(v)}{2} \leq fm(v) \leq m(v).$$

Bizonyítás.

A leghosszabb levélbe vezető úton a feketék száma nem lehet több az élek számánál, hiszen a csúcsot magát nem számoljuk
$\Longrightarrow fm(v) \leq m(v)$. $\surd$
A 6. és 7. pont miatt a leghosszabb úton a pontoknak legalább a fele fekete $\Longrightarrow \frac{m(v)}{2} \leq fm(v)$. $\surd$ $\square$

Tulajdonságok

Állítás

F_v belső csúcsainak száma $b_v \geq 2^{fm(v)} - 1$.

Bizonyítás.

Indukcióval $m(v)$-re: $m(v) = 0 \implies fm(v) = 0$, $b_v \geq 2^0 - 1$ $\surd$

Ha $m(v) > 0$, akkor legyen x, y a két gyereke.

$\implies m(x) < m(v)$ és $m(y) < m(v)$

$fm(v) - 1 \leq fm(x) \leq fm(v)$ és $fm(v) - 1 \leq fm(y) \leq fm(v)$

$b_v = b_x + b_y + 1 \implies$

$b_v \geq (2^{fm(x)} - 1) + (2^{fm(y)} - 1) + 1 \geq 2 \cdot (2^{fm(v)-1} - 1) + 1 = 2^{fm(v)} - 1.$ $\square$

Tulajdonságok

Állítás

Ha egy piros-fekete fában n elemet tárolunk, akkor a fa magassága $\leq 2\log(n+1)$.

Bizonyítás.

Ha r a gyökér $\implies b_r = n$.

$n = b_r \geq 2^{fm(r)} - 1 \implies \log(n+1) \geq fm(r) \geq \frac{m(r)}{2}$ $\surd$ $\square$

Tétel

KERES, MAX, MIN lépésszáma piros-fekete fában $O(\log n)$.

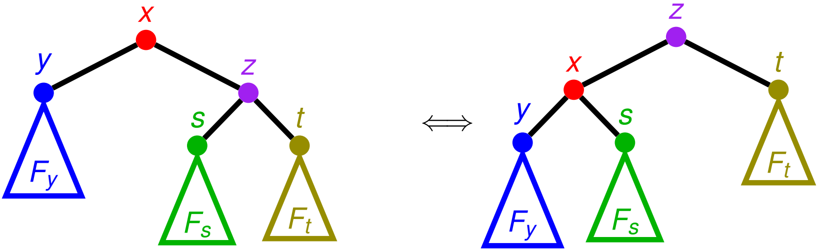
Bizonyítás.

Általában minden keresőfában a lépésszám a fa magasságával arányos $\implies O(\ell) = O(\log n)$. $\square$

BESZÚR lépésszáma

Ha a keresőfáknál használatos beszúrást használnánk, akkor megsérülhetne a piros-fekete tulajdonság.

Forgatás

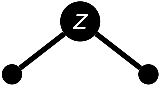


Megj.: Ez a művelet megtartja a keresőfa tulajdonságot.
Lépésszám: $O(1)$

BESZÚR

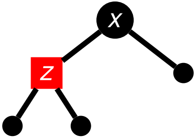
Szúrjuk be az új elemet a keresőfáknál megismert módon. $\Longrightarrow$
Új belső csúcs keletkezik (gyerekei csak üres fekete levelek): z

- Ha z a gyökér, akkor legyen fekete $\Longrightarrow$

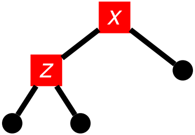


- Ha z nem gyökér, akkor legyen a szülője x, $\Longrightarrow$
  z legyen piros.

  (1) Ha x fekete $\Longrightarrow$ fekete-magasságok sehol nem változnak $\checkmark$ $\Longrightarrow$



  (2) Ha x piros $\Longrightarrow$ nem teljesül a piros-fekete tulajdonság $\Longrightarrow$

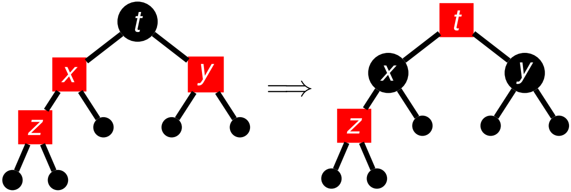


$\Longrightarrow$ további lépések kellenek.

BESZÚR

(2) Mivel x piros, nem gyökér $\Longrightarrow$
legyen x szülője t (fekete), x testvére y.

(2.1) Ha y piros $\Longrightarrow$ átszínezzük t-t pirosra, x, y-t feketére

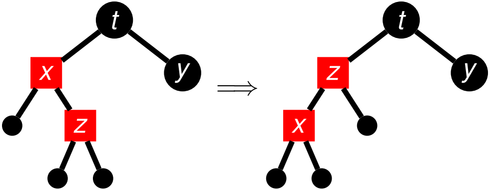


Ha t a gyökér $\Longrightarrow$ t marad fekete $\Longrightarrow$ $fm(t)$ eggyel nagyobb lesz.
Ha t szülője **fekete**, akkor készen is vagyunk, mindenhol teljesül a piros-fekete tulajdonság.
Ha t szülője piros, akkor a problémát két szinttel feljebb toltuk, ott folytatjuk a fa rendbetételét.

BESZÚR

(2.2) Ha y fekete:

(2.2.1) Ha z és x nem azonos oldali gyerek $\Longrightarrow$ forgatunk x körül.

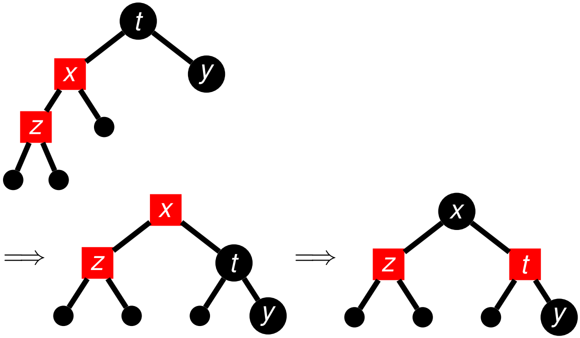


Ezzel a következő esetre vezettük a problémát.

BESZÚR

(2.2) Ha y fekete:

(2.2.2) Ha z és x azonos oldali gyerek
$\Longrightarrow$ forgatunk t körül, majd átszínezünk.



Ezzel ezen részfa gyökerének fekete-magassága nem változik, marad 1, (és az összes ősének fekete magassága sem,) tehát teljesül a piros-fekete tulajdonság. $\surd$

BESZÚR

Tétel

A BESZÚR során

(a) *a lépésszám* $O(\log n)$,

(b) *legfeljebb 2 forgatás történik.*

Bizonyítás.

(a) y piros esetben a (2.1) pontban 2 szinttel feljebb kerül a baj $\Longrightarrow$ szintenként konstans lépés $\Longrightarrow O(\log n)$. $\surd$

(b) Forgatás csak a (2.2) esetben történik, de ekkor nincs felgyűrűzés, rögtön kijavítjuk a fát. $\surd$

$\square$

TÖRÖL

Hasonló módszerek, de bonyolultabb.

Tétel

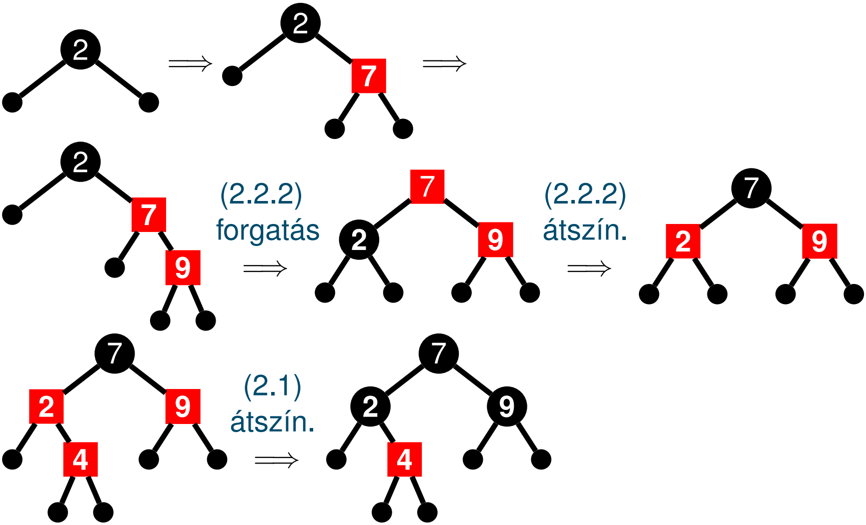
A TÖRÖL során

(a) *a lépésszám* $O(\log n)$*,*

(b) *legfeljebb 3 forgatás történik.*

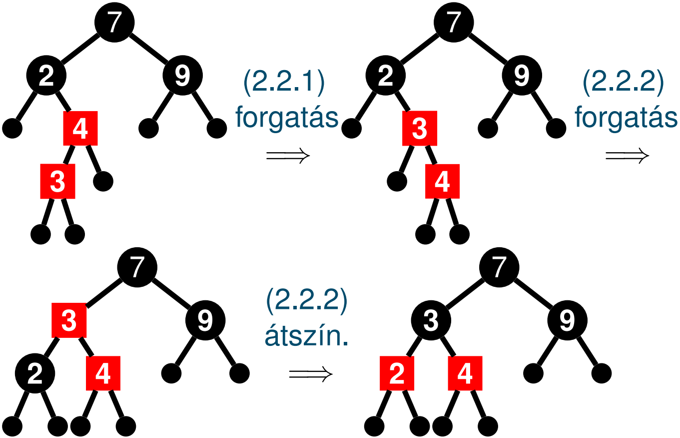
Példa BESZÚRásokra

Szúrjuk be egy üres fába sorban a $2, 7, 9, 4, 3, 1$ elemeket.



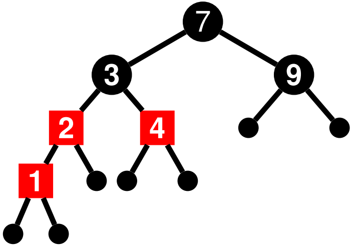
Példa BESZÚRásokra

Szúrjuk be egy üres fába sorban a $2, 7, 9, 4, 3, 1$ elemeket.

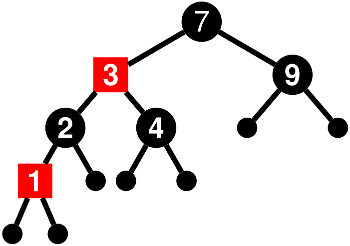


Példa BESZÚRásokra

Szúrjuk be egy üres fába sorban a 2, 7, 9, 4, 3, 1 elemeket.



(2.1)
átszín.
$\Longrightarrow$



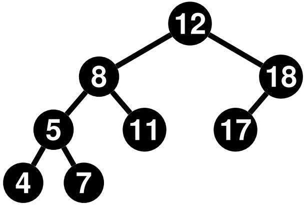
Animáció: Piros-fekete fa

AVL-fák

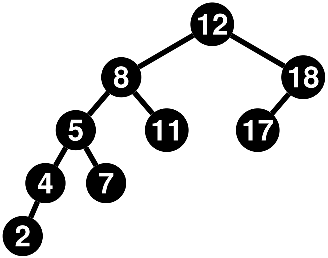
Ez egy másik módja a bináris keresőfák kiegyensúlyozásának.
Gregory Adelson-Velskii és Evgenii Landis, 1962

Definíció (AVL-fa)

Az AVL-fa egy olyan bináris keresőfa, melyben minden csúcsra teljesül, hogy a baloldali részfa magasságának és a jobboldali részfa magasságának különbsége legfeljebb 1.



AVL-fa



nem AVL-fa

AVL-fák

Tétel

Egy n-pontú AVL-fa szintjeinek k száma nem több mint $O(\log n)$, *pontosabban* $k \leq 1,44(\log_2 n + 1)$.

KERES, MIN, MAX, BESZÚR, TÖRÖL lépésszáma: $O(\log n)$

BESZÚR és TÖRÖL műveleteknél az AVL tulajdonság helyreállításakor szükség van egy másik fajta forgatásra.

Animáció: AVL-fa