Algoritmusok és adatszerkezetek II.

2-3-4 fák és piros-fekete fák

Szegedi Tudományegyetem

Emlékeztető

Coospace kvíz

Az első Coospace kvíz ma 20:00-tól tölthető ki.
Végső kitöltési határidő: április 9. 10:00

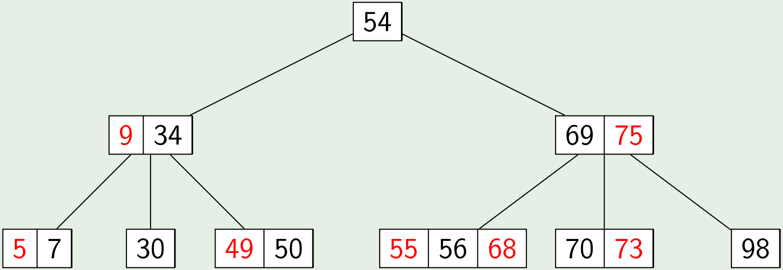
Egyebek

- Hallgatói ösztöndíj a mestint tanszéken
- Szorgalmi feladatok
- Értékelés

Definíció

2-3-4 fa alatt olyan **általános keresőfát** értünk, amelynek minden x csúcsára $Rang(x) \in \{1, 2, 3\}$ (Mégis miért hívjuk akkor 2-3-4 fának?)

Példa



Piros-fekete fák tulajdonságai

1. Minden csúcs színe piros vagy fekete
2. A gyökér színe fekete
3. Minden levele[1] fekete
4. A piros csúcsoknak **kizárólag** fekete színű gyerekeik vannak
5. Bármely csúcsból azonos számú fekete csúcs érintésével jutunk el bármelyik levélbe

[1] levelek alatt itt most az "őrszemeket" értjük

Piros-fekete fák tulajdonságai

1. Minden csúcs színe piros vagy fekete
2. A gyökér színe fekete
3. Minden levele[1] fekete
4. A piros csúcsoknak **kizárólag** fekete színű gyerekeik vannak
5. Bármely csúcsból azonos számú fekete csúcs érintésével jutunk el bármelyik levélbe

Tétel

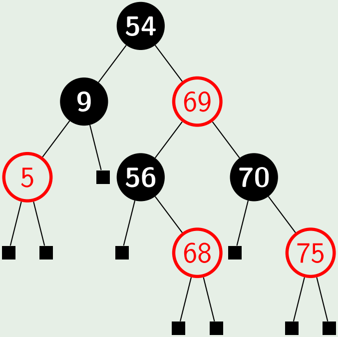
Bármely n kulcsú piros-fekete fa magassága legfeljebb $2\log(n+1)$.

[1] levelek alatt itt most az "őrszemeket" értjük

Fekete-magasság

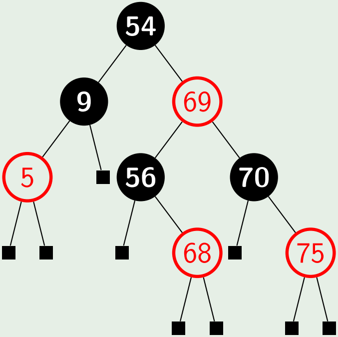
- $bh(x)$ jelölje az x csúcsból induló, bármely levélig vezető úton található, (x-en kívüli) fekete csúcsok számát

Példa



- $bh(x)$ jelölje az x csúcsból induló, bármely levélig vezető úton található, (x-en kívüli) fekete csúcsok számát

Példa



A (teljes) fa fekete-magassága 2
A 9 gyökerű fa fekete-magassága 1
A 69 gyökerű fa fekete-magassága 2

A piros-fekete fák legrosszabb magasságának igazolása

Fontos észrevételek

1. x gyökerű fa $h(x)$ magassága $\geq bh(x)$
2. x minden y gyerekére $bh(y) = bh(x)$ vagy $bh(y) = bh(x) - 1$
3. x minden y gyerekére $h(y) = h(x) - 1$ (másképp $h(y) < h(x)$)

A piros-fekete fák legrosszabb magasságának igazolása

Fontos észrevételek

1. x gyökerű fa $h(x)$ magassága $\geq bh(x)$
2. x minden y gyerekére $bh(y) = bh(x)$ vagy $bh(y) = bh(x) - 1$
3. x minden y gyerekére $h(y) = h(x) - 1$ (másképp $h(y) < h(x)$)

Minden x gyökerű fa legalább $2^{bh(x)} - 1$ csúcsot tartalmaz.

0 magas fára természetesen teljesül.
Az x gyökerű fában legalább annyi csúcs van, mint ahány csúcs a fiaiban legalább van $+$ 1, azaz $2 * (2^{bh(x)-1} - 1) + 1 = 2^{bh(x)} - 1$

A piros-fekete fák legrosszabb magasságának igazolása

Fontos észrevételek

1. x gyökerű fa $h(x)$ magassága $\geq bh(x)$
2. x minden y gyerekére $bh(y) = bh(x)$ vagy $bh(y) = bh(x) - 1$
3. x minden y gyerekére $h(y) = h(x) - 1$ (másképp $h(y) < h(x)$)

Minden x gyökerű fa legalább $2^{bh(x)} - 1$ csúcsot tartalmaz.

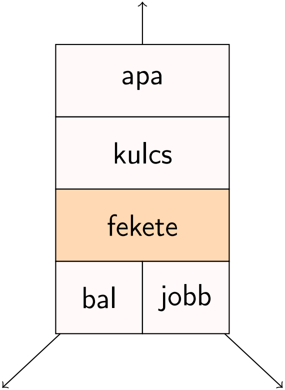
0 magas fára természetesen teljesül.
Az x gyökerű fában legalább annyi csúcs van, mint ahány csúcs a fiaiban legalább van + 1, azaz $2 * (2^{bh(x)-1} - 1) + 1 = 2^{bh(x)} - 1$

A piros-fekete fák 4. tulajdonságából következően

Bármely x-ből levélig menő úton az érintett csúcsok **legalább** 1/2-e fekete $\Rightarrow bh(x) \geq h(x)/2$, vagyis az x gyökerű fában lévő n kulcsok száma $n \geq 2^{h/2} - 1 \Rightarrow h \leq 2\log(n+1)$

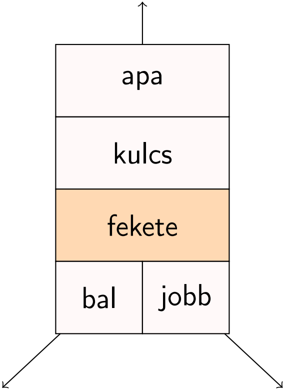
Piros-fekete fa implementációja

```
class Node {
    Object kulcs;
    boolean fekete;
    Node *apa;
    Node *bal;
    Node *jobb;
}
```



Piros-fekete fa implementációja

```
class Node {
    Object kulcs;
    boolean fekete;
    Node *apa;
    Node *bal;
    Node *jobb;
}
```



Megjegyzés

Az eddigi kiegészítő információk közül a legolcsóbb (csupán 1 bit)

AVL vs. piros-fekete fa

AVL fa legrosszabb magassága jobb, mint a piros-fekete fáé

$h < 1.45 * h_{OTP}(n)$ vs. $h \leq 2 * h_{OPT}(n)$

- $h_{OPT}(n)$ az n kulcsot tartalmazó, teljesen kiegyensúlyozott bináris keresőfa magassága

AVL vs. piros-fekete fa

AVL fa legrosszabb magassága jobb, mint a piros-fekete fáé

$h < 1.45 * h_{OTP}(n)$ vs. $h \leq 2 * h_{OPT}(n)$

- $h_{OPT}(n)$ az n kulcsot tartalmazó, teljesen kiegyensúlyozott bináris keresőfa magassága
- Gyakorlati jelentősége minimális: $n = 2^{30} - 1 > 10^9$-ra azt kapjuk, hogy egy AVL/piros-fekete-fa legfeljebb 44/60 magas
- Ha a Keres művelet végrehajtása dominál, el lehet gondolkozni az AVL-fa használatán

AVL vs. piros-fekete fa

AVL fa legrosszabb magassága jobb, mint a piros-fekete fáé

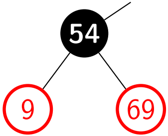
$h < 1.45 * h_{OTP}(n)$ vs. $h \leq 2 * h_{OPT}(n)$

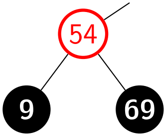
- $h_{OPT}(n)$ az n kulcsot tartalmazó, teljesen kiegyensúlyozott bináris keresőfa magassága
- Gyakorlati jelentősége minimális: $n = 2^{30} - 1 > 10^9$-ra azt kapjuk, hogy egy AVL/piros-fekete-fa legfeljebb 44/60 magas
- Ha a Keres művelet végrehajtása dominál, el lehet gondolkozni az AVL-fa használatán

- A Beszúr és Töröl műveletek ugyanakkor implementációs szempontból egyszerűbbek/gyorsabbak piros-fekete fákra

A helyreállítás (egyik) záloga – átszínezés

- Ha van egy valid piros-fekete fánk, amelynek egy x csúcsa fekete, fiai viszont pirosak, akkor az 5. tulajdonságot nem sértő fát kapunk akkor is, ha x pirosra, fiai pedig feketére váltanak
- Másképpen, ha piros testvérpár színe feketére vált, a közös szülőnek a továbbiakban nem kell feketének lennie



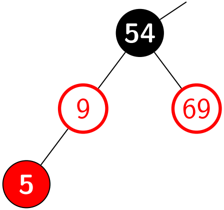


Beszúrás utáni javítás átszínezéssel

- A beszúrandó kulcsot piros színnel szúrjuk be (arra a helyre ahova egyébként egy bináris keresőfába tennénk)
- Mitől romolhat el beszúrás kapcsán a piros-fekete fa?

Beszúrás utáni javítás átszínezéssel

- A beszúrandó kulcsot piros színnel szúrjuk be (arra a helyre ahova egyébként egy bináris keresőfába tennénk)
- Mitől romolhat el beszúrás kapcsán a piros-fekete fa?
  - Ha piros szülő jut piros gyerekhez



Beszúrás utáni javítás átszínezéssel

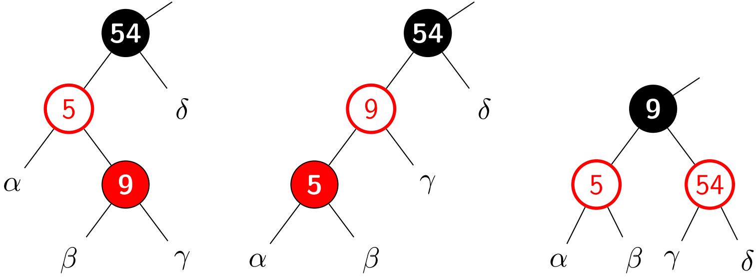
- A beszúrandó kulcsot piros színnel szúrjuk be (arra a helyre ahova egyébként egy bináris keresőfába tennénk)
- Mitől romolhat el beszúrás kapcsán a piros-fekete fa?
  - Ha piros szülő jut piros gyerekhez

Mi lehet ennek a helyreállításnak a velejárója?

A pirosra színezett csúcs mentén újabb helyreállításra lehet szükség

Beszúrás utáni javítás forgatással

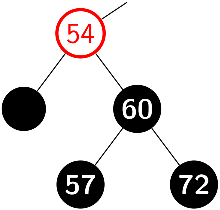
- $\alpha, \beta, \gamma, \delta$ egy-egy részfát jelöl
- Az előző helyzettől az különbözteti meg a mostanit, hogy a beszúrt kulcs nagybátyja (a δ részfa gyökere) fekete
- AVL fákhoz hasonlóan, itt is a cikk-cakkos esetben van szükség 2 forgatásra (a cikk-cakkot δ-hoz viszonyítva nézzük)



(a) 2 forgatás kell (b) 1 forgatás elég (c) helyreállítás után

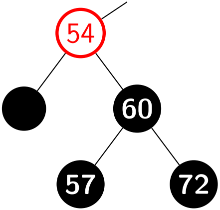
Törlés utáni javítás átszínezéssel

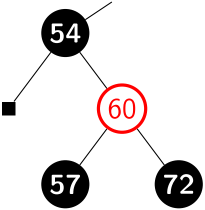
- Bajt az okozhat, ha kiesik egy fekete csúcs
- A kieső csúcs feketeségét át kell ruházni, duplán fekete csúcsunk viszont nem maradhat
- Ha a duplán fekete csúcs testvére és unokaöccsei is feketék, a nagyszülő át tudja vállalni a feketeséget



Törlés utáni javítás átszínezéssel

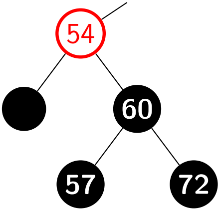
- Bajt az okozhat, ha kiesik egy fekete csúcs
- A kieső csúcs feketeségét át kell ruházni, duplán fekete csúcsunk viszont nem maradhat
- Ha a duplán fekete csúcs testvére és unokaöccsei is feketék, a nagyszülő át tudja vállalni a feketeséget

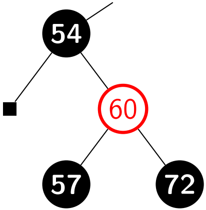




Törlés utáni javítás átszínezéssel

- Bajt az okozhat, ha kiesik egy fekete csúcs
- A kieső csúcs feketeségét át kell ruházni, duplán fekete csúcsunk viszont nem maradhat
- Ha a duplán fekete csúcs testvére és unokaöccsei is feketék, a nagyszülő át tudja vállalni a feketeséget



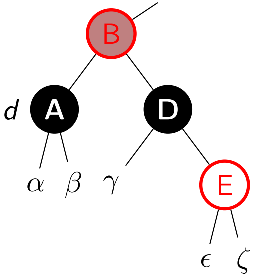


Mi történt volna, ha a duplán fekete csúcs apja nem piros?

Szükség esetén a helyreállítást az apától folytathatjuk tovább.

Törlés utáni javítás forgatással

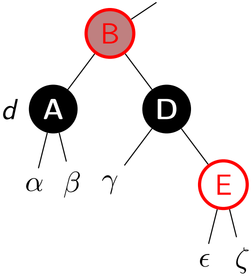
- Piros unokaöcs esetén forgatásra van szükség
- Ha a (*d*-vel megjelölt) duplán fekete csúcs távolabbi unokaöccse (E) piros, 1 forgatás is elég



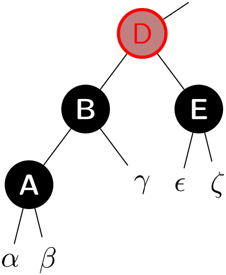
(a) helyreállítás előtt

Törlés utáni javítás forgatással

- Piros unokaöcs esetén forgatásra van szükség
- Ha a (d-vel megjelölt) duplán fekete csúcs távolabbi unokaöccse (E) piros, 1 forgatás is elég
- Valóban, az egyes részfákig a forgatást követően is ugyanannyi fekete csúcs érintésével tudunk eljutni
  - α és β: 2/3; γ, ϵ és ζ: 1/2 (B kezdeti színétől függően)



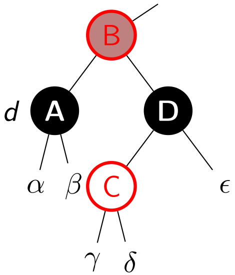
(a) helyreállítás előtt



(b) helyreállítás után

Törlés utáni javítás segédforgatással

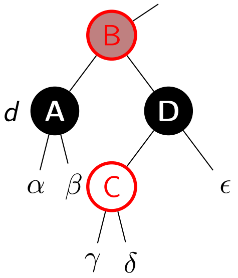
- Piros unokaöcs esetén forgatásra van szükség
- Ha a (d-vel megjelölt) duplán fekete csúcs közelebbi unokaöccse piros, segédforgatásra van szükség



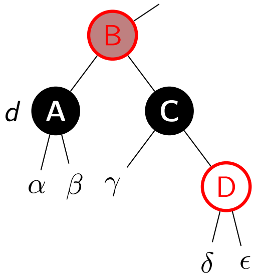
(a) segédforgatás előtt

Törlés utáni javítás segédforgatással

- Piros unokaöcs esetén forgatásra van szükség
- Ha a (d-vel megjelölt) duplán fekete csúcs közelebbi unokaöccse piros, segédforgatásra van szükség
- 1 további forgatással garantáltan orvosolható (hiszen a duplán fekete csúcs távolabbi unokaöccse piros lett)



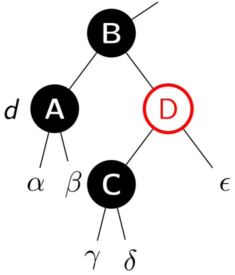
(a) segédforgatás előtt



(b) segédforgatás után

Egy további eshetőség segédforgatásra

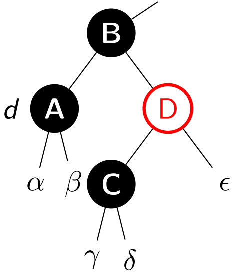
- Piros testvér esetén is forgatni kell



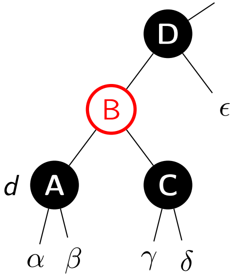
(a) segédforgatás előtt

Egy további eshetőség segédforgatásra

- Piros testvér esetén is forgatni kell
- A duplán fekete csúcs megmarad, de a forgatást követően biztosan valamelyik már tárgyalt eset áll elő

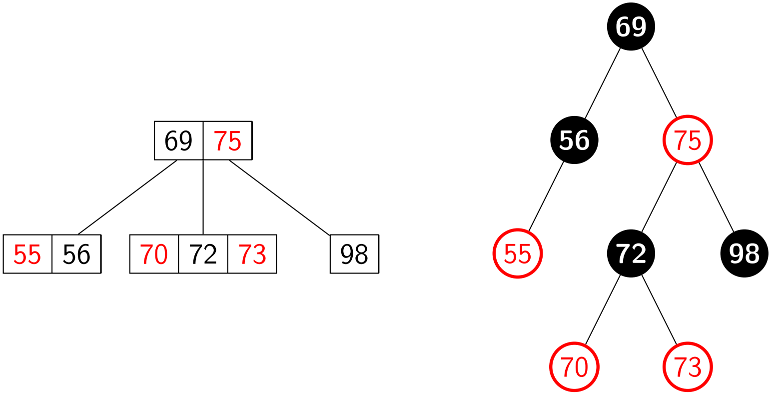


(a) segédforgatás előtt



(b) segédforgatás után

2-3-4 és piros-fekete fák közötti kapcsolat



A fekete csúcsokat piros leszármazottaikkal összeolvasztva 2-3-4 fát kapunk.

Összegzés

- A piros-fekete fák az AVL fákhoz hasonló tulajdonságúak
- Piros-fekete fákkal tulajdonképp egy speciális hatékony általános keresőfát (2-3-4 fa) valósítunk meg