

9. tétel

Adatszerkezetek és algoritmusok vizsga

Frissült: 2013. január 28.

Piros-fekete fák

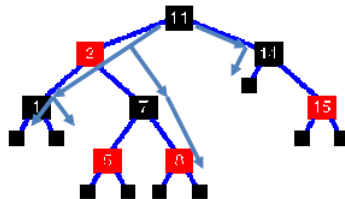
A piros-fekete fa olyan bináris keresőfa, melynek minden pontja egy extra bit információt hordoz: ez a pont színe, amelynek értékei: PIROS vagy FEKETE.

Az ilyen fák közelítőleg kiegyensúlyozottak: a piros-fekete fában bármely, a gyökértől levélig vezető út hossza nem lehet nagyobb, mint a legrövidebb ilyen út hosszának kétszerese.

A fa minden pontja tartalmazza a szín, kulcs, bal, jobb és szülő mezőket. A levelek speciális, NIL-t tartalmazó elemek lesznek, a fa kulcsot tartalmazó pontjai a belső pontok.

Egy piros fekete fa

- Minden csúcsa PIROS vagy FEKETE,
- a gyökér, és minden levél FEKETE,
- ha egy csúcs PIROS, akkor mindkét gyereke FEKETE.
- minden csúcsból minden út egy levélig ugyanannyi FEKETE csúcsot tartalmaz



A piros-fekete fák keresési ideje: $O(\log_2 n)$.

Műveletek

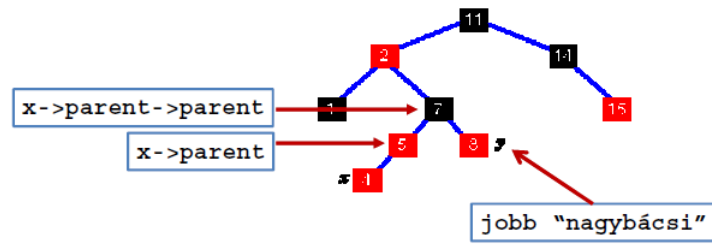
Beszúrás

Bináris keresőfához hasonlóan szűrjük be az elemet, amit rögtön pirosra is színezzünk.

Helyre kell állítani a Piros-Fekete fa tulajdonságát.

Addig vizsgálódunk, amíg nem értük el a gyökeret és a szülő (aktuális elem, x szülője) piros. Most x szülője legyen balra a nagyszülőjétől.

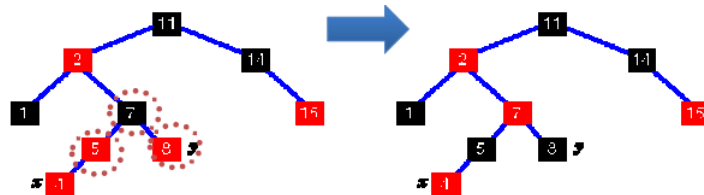
Legyen ez a beszúrt elem az x , a szülője $x \rightarrow \text{parent}$, nagyszülője $x \rightarrow \text{parent} \rightarrow \text{parent}$, jobb nagybátyját is figyeljük (y).



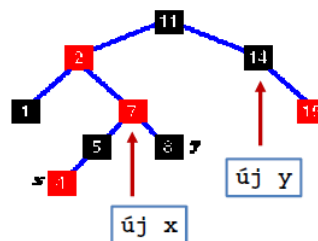
Több eset lehet:

1. Szülő PIROS, Nagybácsi PIROS, nagyszülő FEKETE.

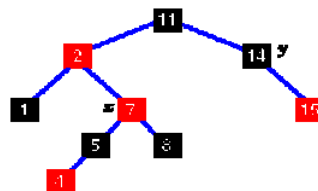
Megcseréljük y , nagyszülő és szülő színét.



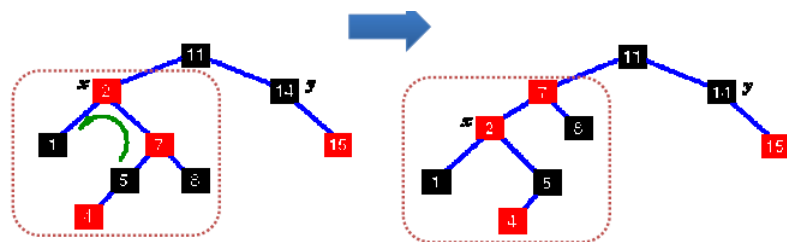
x új értéke legyen a nagyszülő, x szülője megint balgyerek, jelöljük y -nal x nagybácsiját, de ez most fekete



2. szülő PIROS, nagybácsi FEKETE, és x jobb gyerek.

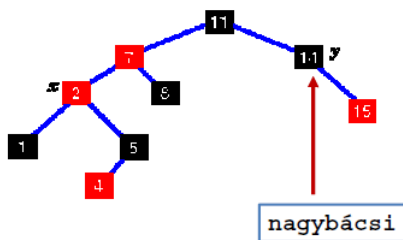


Felvisszük x -et és balra forgatunk x körül.

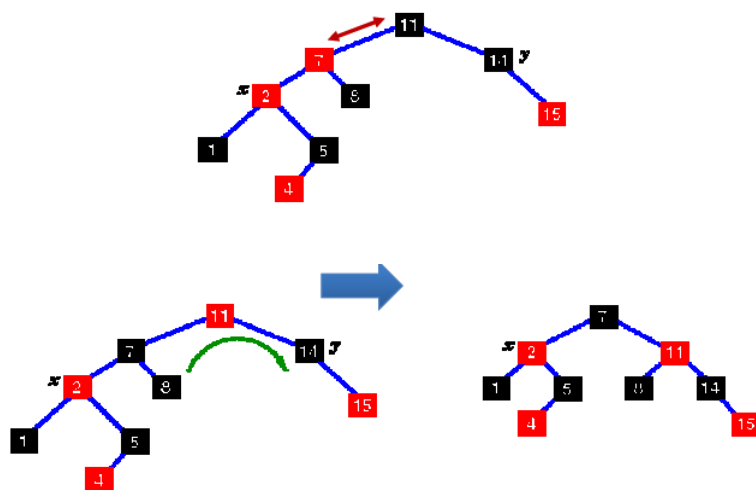


De ekkor még x szülője piros lesz.

3. szülő PIROS, nagybácsi FEKETE, x balgyerek.



Kicseréljük a színeket és forgatunk:



Ez már egy jó PF fa!

Ekvivalens esetszétválasztás, amikor a szülő a nagyszülő jobbán van!

Mindig addig megyünk felfelé, amíg az x szülője is piros.

Pszeudokód.

```
Fába-beszúr(T,x) -- beszúrom a bináris keresőfába
szín[x] ← PIROS
while x ≠ gyökér[T] and szín[szülő[x]] = PIROS
do if szülő[x] = bal[szülő[szülő[x]]]
    then y ← jobb[szülő[szülő[x]]]
    if szín[y] = PIROS
    then szín[szülő[x]] ← FEKETE --1.
        szín[y] ← FEKETE --1.
        szín[szülő[szülő[x]]] ← PIROS --1.
        x ← szülő[szülő[x]] --1.
    else if x = jobb[szülő[x]]
    then x ← szülő[x] --2.
        BALRA-FORGAT(T,x) --2.
        szín[szülő[x]] ← FEKETE --3.
        szín[szülő[szülő[x]]] ← PIROS --3.
        JOBBRA-FORGAT(T, szülő[szülő[x]]) --3.
else ua., mint a then, csak a bal és jobb felcserélve
szín[gyökér[T]] ← FEKETE
```

Törlés

Törlés alapja a bináris keresőfában törlés.

Ha piros elemet törlünk, megmarad a PF tulajdonság. Probléma a fekete elem törlésénél lehet.

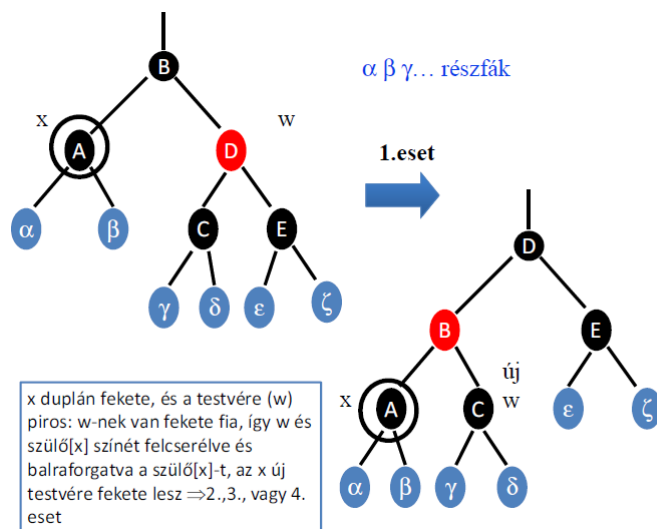
Ahhoz, hogy helyreállítsuk a fekete csúcsok számát minden úton, a gyerekeknek adjuk a kitörölt csúcs fekete színét.

Úgy képzeljük, hogy a gyerekének most van egy extra feketéje:

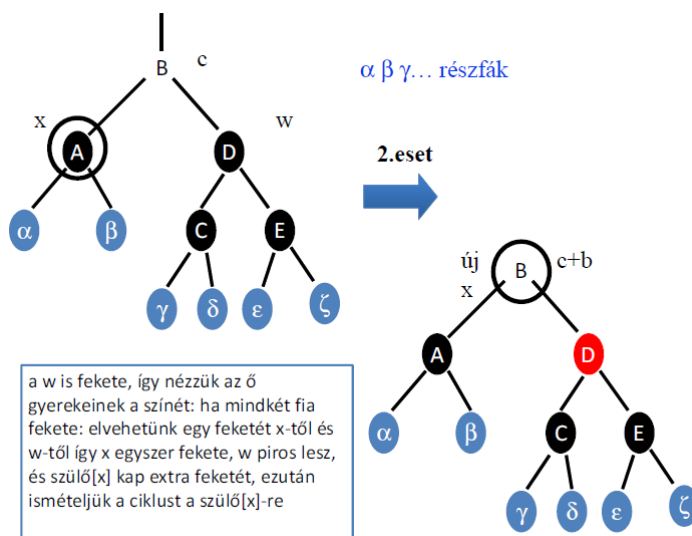
- Ha a gyerek színe FEKETE, akkor most duplafekete
- Ha a gyerek színe PIROS, akkor most piros-fekete.

Lehetséges esetek. Csak azokat vesszük figyelembe, ahol x balgyerek, ahol jobb, az szimmetrikusan adódik. (w jelöli x testvérét ($w \leftarrow \text{jobb[szülő[x]]}$))

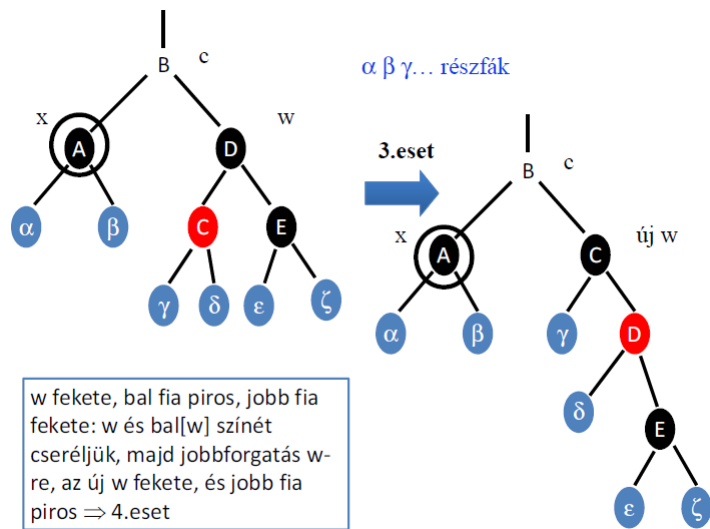
1. x testvére w piros



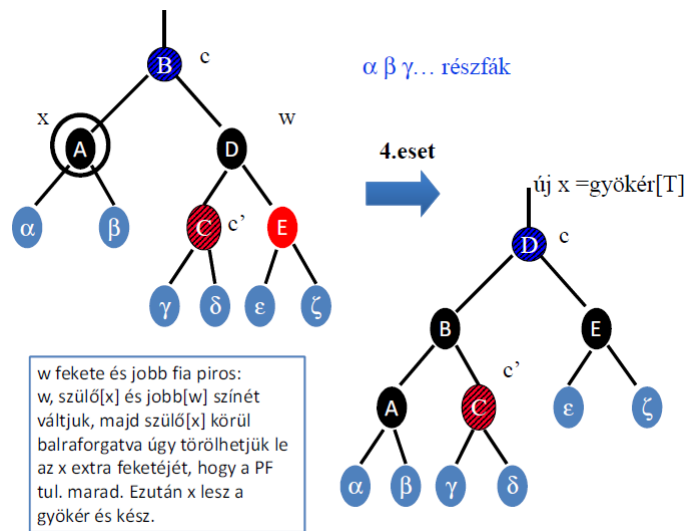
2. x testvére w fekete és w mindkét gyereke fekete



3. x testvére w fekete, w balgyereke piros, jobbgyereke fekete



4. x testvére w fekete, w jobbgyereke piros



Pszudokód:

PF-FÁBÓL-TÖRÖL-JAVÍT(T,x)

```

while x  $\neq$  gyökér[T] and szín[x] FEKETE
do if x = bal[szülő[x]]
then w  $\leftarrow$  jobb[szülő[x]]
    if szín[w]=PIROS
    then szín[w]  $\leftarrow$  FEKETE; szín[szülő[x]]  $\leftarrow$  PIROS
        BALRA-FORGAT(T,szülő[x]); w  $\leftarrow$  jobb[szülő[x]]
    if szín[bal[w]] = FEKETE and szín[jobb[w]] = FEKETE
    then szín[w]  $\leftarrow$  PIROS; x  $\leftarrow$  szülő[x]

```

```

        else if szín[jobb[w]] = FEKETE
            then szín[bal[w]] ← FEKETE; szín[w] ← PIROS
                JOBBRA-FORGAT(T,w); w ← jobb[szülő[x]]
                szín[w] ← szín[szülő[x]]
                szín[szülő[x]] ← FEKETE;
        szín[jobb[w]] ← FEKETE
            BALRA-FORGAT(T, szülő[x])
            x ← gyökér[T]
        else ua., mint a then, csak a bal és jobb felcserélve
        szín[x] ← FEKETE

```