Piros-fekete fák

(2007. április 15.)

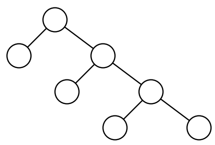
A *piros-fekete fa* egy bináris fa, amiben

1. minden nem levél(=belső) csúcsnak két fia van
2. elemeket a belső csúcsokban tárolunk csak (levélben nem)
3. teljesül a keresőfa tulajdonság
4. a fa minden csúcsa piros vagy fekete
5. a gyökér fekete
6. a levelek feketék
7. piros csúcs mindkét gyereke fekete
8. minden v csúcsra igaz, hogy az összes v-ből levélbe vezető úton ugyanannyi fekete csúcs van.

1. Megjegyzés. Az 1-2 tulajdonságok technikai feltételek. Úgy is szokás fogalmazni, hogy veszünk egy szokásos F bináris keresőfát és mindenhol, ahol F-ben hiányzik egy ág (mert vagy csak egy fia van a csúcsnak, vagy mert levél), ott kiegészítjük a fát és keletkezik egy új levél amiben nem lesz kulcs (F leveleinél mindkét hiányzó ágat behúzzuk, azaz 2 új levél keletkezik az egy régi helyett). Tehát a fának csak új típusú levelei lesznek, minden eredeti pont belső. Igazából elég egyetlen új csúcsot hozzávenni F-hez, és abba vezetni minden sehova nem mutató mutatót (bár ez már akkor nem fa, de az eljárásainkat ez nem fogja zavarni). Sokszor csak az eredeti F fát rajzolják piros-fekete faként.

Jelölések:
F_v jelöli a v gyökerű részfát.
$m(v)$ = a v csúcs magassága, azaz hogy hány élből áll a v-ből levélbe vezető leghosszabb út (csak lefelé megyünk, ahogy egy keresőfánál illik). Tehát pl. levélre $m(v) = 0$.
$fm(v)$ = a v csúcs fekete magassága, azaz hogy a v-ből levélbe vezető utakon v nélkül hány fekete csúcsot látunk. (Ez a 8. tulajdonság miatt minden úton egyforma.)

Feladat. Lehet-e egy piros-fekete fa alakja ilyen?



Megoldás. Nem, mert a bal oldali ág miatt a gyökérre $fm(r) = 1$, és így a gyökéren és a leveleken kívül nincs fekete csúcs, ami 7. miatt nem lehetséges.

1. Állítás. *Egy piros-fekete fa minden v csúcsára teljesül, hogy*

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

Bizonyítás. Nyilván $fm(v) \leq m(v)$ mindig igaz. Egy úton a 7. tulajdonság miatt nem lehet két egymás utáni piros csúcs, tehát a csúcsok legalább fele fekete.

2. Állítás. *Egy piros-fekete fában az F_v részfa belső csúcsainak száma legalább $2^{fm(v)} - 1$.*

Bizonyítás. Jelölje b_v az F_v részfa belső csúcsainak számát. $m(v)$ szerinti teljes indukcióval bizonyítunk: Ha $m(v) = 0$, akkor v levél, $fm(v) = 0$, és $b_v = 0$.

Ha $m(v) > 0$, akkor v egy belső csúcs, aminek 2 fia van, legyenek ezek x és y. Világos, hogy $m(x), m(y) < m(v)$ és $fm(v) - 1 \le fm(x), fm(y) \le fm(v)$, ezért az indukciós feltevés szerint

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

3. Tétel. *Ha egy piros-fekete fában n elemet tárolunk, akkor a fa magassága legfeljebb $2 \cdot \log(n+1)$.*

Bizonyítás. Legyen r a fa gyökere. Ha n elemet tárolunk, akkor a belső csúcsok száma $b_r = n$. Az előző állítás szerint $b_r \ge 2^{fm(r)} - 1$. Ebből $\log(n+1) \ge fm(r)$. Ha felhasználjuk, hogy $fm(r) \ge m(r)/2$ (lásd 1. Állítás), akkor a kívánt becslést kapjuk.

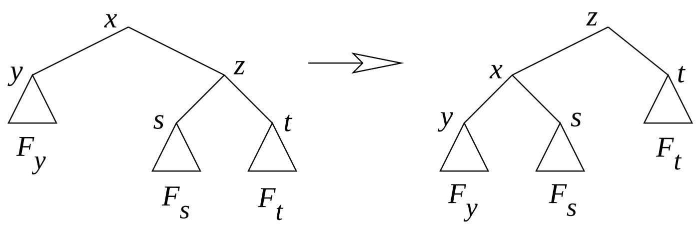
Ebből következik az alábbi

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

A BESZÚR és TÖRÖL eljárásoknál cél, hogy lehetőleg helyi kis változtatásokkal biztosítsuk a piros-fekete fa tulajdonságokat. Ez forgatásokkal, és átszínezésekkel megy, kivéve 1-1 esetet, amikor a problémát a fában feljebbi szintre toljuk. A következőkben az ábrákon látható részfákban karika jelzi a fekete csúcsokat, négyzet a pirosat és nincs semmi, ha nem tudjuk/mindegy hogy milyen színű a csúcs. A szimetrikus esetekből csak egyet rajzolok fel.

Forgatások

Egy bináris fában (nem csak piros-fekete fában) tetszőleges csúcsnál végezhetünk balra vagy jobbra forgatást. Az x-nél *balra forgatás* az F_x-en kívüli csúcsok helyét nem változtatja, F_x csúcsait pedig az alábbiak szerint rendezi át:



A jobb oldali fából z-nél *jobbra forgatás*sal visszakapjuk a baloldali fát.

2. Megjegyzés. Vegyük észre, hogy ha az eredeti bináris fa rendelkezett a keresőfa tulajdonsággal, akkor a forgatás után is keresőfát kapunk, de a piros-fekete tulajdonság elromolhat.

A BESZÚR eljárás

Végezzük el a belső csúcsok által alkotott bináris fába való naív beszúrást (ekkor a piros-fekete fogalmaival új belső csúcs keletkezik). Legyen ez z.

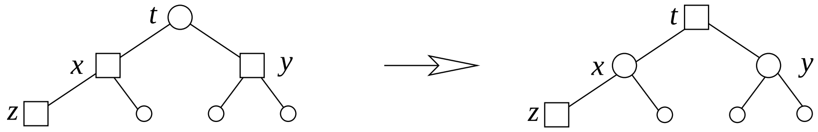
– Ha z gyökér (azaz a fa első csúcsa), akkor z legyen fekete.

– Ha nem gyökér, akkor az apját jelölje x. Kezdetben legyen z piros.

(1) Ha x **fekete**, akkor a fa piros-fekete fa maradt – készen vagyunk.

(2) Ha x **piros**, akkor sérül a 7. tulajdonság. Ilyenkor x nem a gyökér (mert a gyökér fekete). Jelölje x testvérét y, apját t. Mivel x piros, ezért t fekete (7. tulajdonság)

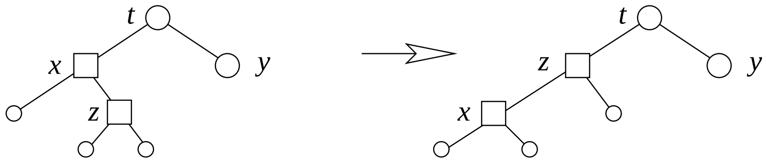
(2.1) Ha y **piros**, akkor átszínezünk: x és y fekete lesz, t pedig piros. Így a problémát két szinttel feljebb toltuk.



Ha t a gyökér, akkor t színét hagyjuk meg feketének – ebben az esetben a fa fekete magassága 1-gyel nő.

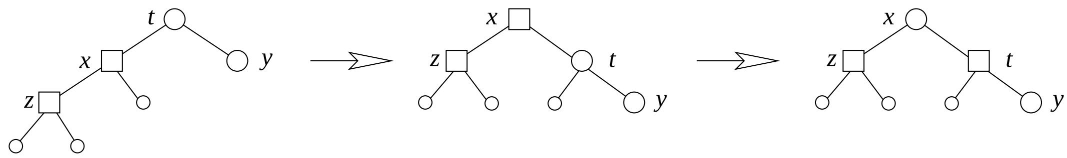
(2.2) Ha y **fekete**, akkor forgatni is kell:

(2.2.1) ha z és x nem azonos oldali gyerekek, akkor forgassunk x-nél, hogy egy oldalra kerüljenek:



Ezzel visszavezettük a következő esetre (csak x és z fel van cserélve).

(2.2.2) ha azonos oldaliak, akkor egy szimpla forgatás után átszínezünk úgy, hogy az egyes ágak fekete magassága az eredetihez képest ne változzon:



5. Tétel. *A BESZÚR eljárás során*
(a) a lépésszám $O(\log n)$,
(b) legfeljebb 2 forgatás történik.

Bizonyítás. (a) Az y piros esetben az átszínezéskor egy szinttel feljebb csinálhatunk bajt, ez felgyűrűzhet a fa tetejéig. Minden más esetben viszont rögtön készen vagyunk.
(b) Forgatás csak az y fekete esetben történik, akkor egyből helyre is állítjuk a fát.

3. Megjegyzés. Az y=fekete első esetében két egymás utáni forgatást fogunk végrehajtani.Ezt akár egyben megcsinálhatjuk, és akkor már csak a színezést kell a leírtak szerint megváltoztatni.

A TÖRÖL eljárás

A belső csúcsokból álló bináris fában hajtsuk végre a naív törlést úgy, hogy a csúcsok színén nem változtatunk akkor sem, ha esetleg másik kulcs került a csúcsba. Legyen y az a csúcsa a fának, ami ennél a törlésnél ténylegesen megszűnik (ez nem feltétlenül egyezik meg azzal a csúccsal, ahol a törlendő érték volt). Mivel y-nak a belső csúcsok által alkotott fában legfeljebb 1 fia van, y a piros-fekete fának egy olyan belső csúcsa, aminek legalább az egyik fia levél. Legyen x a nem levél fia, ha ilyen van, ha nincs, akkor valamelyik fia.

– Ha y **piros**, akkor elhagyjuk és x-et rakjuk a helyébe. (A másik fiú eltünik, x megőrzi a színét.)

– Ha y **fekete**, akkor elhagyása 1-gyel csökkenti a részfában az fm értékeket — hogy ezt ellensúlyozzuk, x kap egy „fekete pontot".

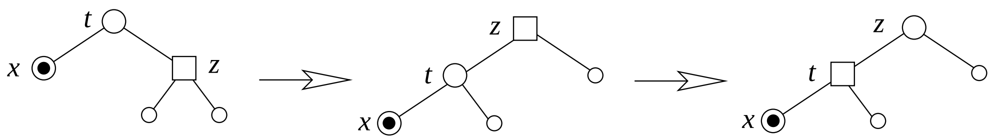
– Általában, ha egy fekete ponttal rendelkező csúcs piros, akkor változtassuk a színét feketére és ezzel a fekete pontot felhasználtuk.

– Ha egy fekete pontos x csúcs fekete, több eset van:

(1) x a gyökér: a fekete pontot elfelejtjük. Ez az az eset, amikor a fa fekete mélysége csökken.

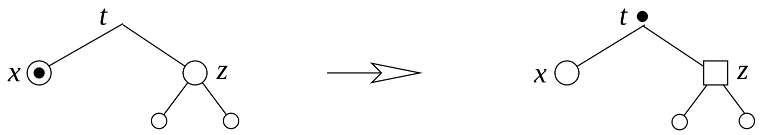
(2) x nem a gyökér, akkor van apja, legyen ez t, az x testvérét jelölje z

(2.1) Ha z **piros**, akkor t fekete és egy forgatással+átszínezéssel elérhetjük, hogy x új testvére fekete legyen, de a részfák fekete magassága ne változzék. Innen a következő, (2.2) eset szerint járunk el.

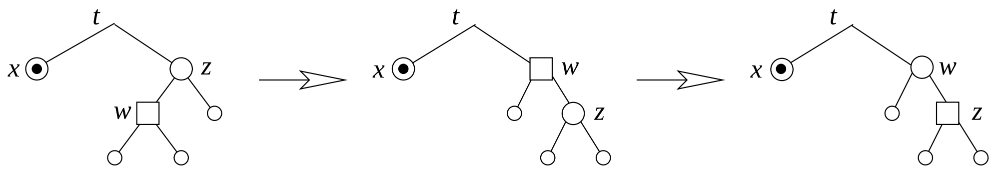


(2.2) z **fekete**. Vegyük észre, hogy z nem lehet levél, hiszen a fekete pont miatt eredetileg az x gyökerű részfának legalább 1 kellett legyen a fekete magassága.

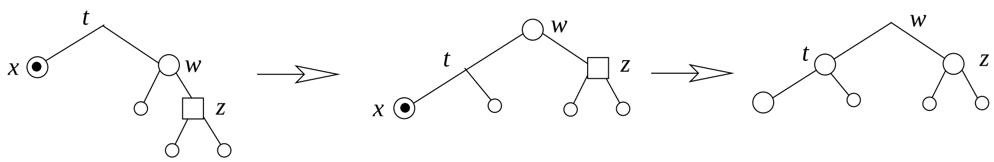
(2.2.1) Ha z fiai feketék, akkor legyen z piros és a fekete pontot adjuk x helyett az apának t-nek.



(2.2.2) Ha z fiai között van piros, akkor egy forgatással z-nél és a fekete magasság helyreállításával elérhetjük, hogy az x-től „távolabbi" fia z-nek (azaz az x-szel ellentétes oldali fiú) piros legyen.



Ebben az esetben pedig t-nél forgatunk és felhasználjuk a fekete pontot:



A végén w színe a t eredeti színe lesz.

6. Tétel. *A TÖRÖL eljárás során*
(a) a lépésszám $O(\log n)$,
(b) legfeljebb 3 forgatás történik

Bizonyítás. (a) Csak amikor z fekete és a fiai is feketék, akkor nem elég a lokális változtatás, de ilyenkor a fekete pont csak felfelé vándorolhat a fán.
(b) Ha forgatunk (ebből bármely esetben max három kell), akkor kész is vagyunk.

További információk

- wikipedia: http://en.wikipedia.org/wiki/Red_black_tree
- animációval: http://www.ececs.uc.edu/˜ franco/C321/html/RedBlack/
- Cormen, Leiserson, Rivest, Stein: Új algoritmusok