Algel VII. gyakorlat

Keresünk, és még mindig fázunk

2011. március 21.

5. **Az $[1, 178]$ intervallum összes egészei egy 2-3 fában helyezkednek el. Tudjuk, hogy a gyökérben két kulcs van, és ezek közül az első 17. Mi lehet a második? Miért?**
Ekkor a bal részfában legfeljebb 16 elemet tárolhatunk, tehát legfeljebb 5 szintje lehet. Mivel minden levél egy szinten helyezkedik el, ezért a középső és jobb oldali részfa is legfeljebb 5 szintű lehet. Ebben az esetben itt 81 és 81 elemet tárolhatunk legfeljebb. Ezeket összeadva kiderül, hogy ennyit muszáj is, hiszen csak így lehet 178 elemet tárolni. Ekkor viszont a második kulcs a gyökérben $16 + 81 + 1 = 98$ lesz.

6. **Az S_1 és S_2 kulcshalmazokat kiegészített 2-3 fákban tároljuk. Ezek az eredeti 2-3 fától annyiban különböznek csak, hogy minden csúcsban nyilván van tartva az onnan induló részfa magassága. Tudjuk továbbá, hogy az S_1-beli kulcsok mind kisebbek, mint az S_2-beliek. Javasoljunk hatékony algoritmust a két fa egyesítésére!**
Legyenek a magasságok m_1 és m_2! Az általánosság megsértése nélkül legyen $m_1 \leq m_2$. Ekkor (mivel minden kulcs az egyikben kisebb, mint a másikban) a levelek nem fognak átlapolódni. Ezért egyszerűen az $m_2 - m_1$-edik szinten beszúrjuk a szokásos algoritmussal a kisebbik fa gyökerét, ami $O(m_2 - m_1)$ lépésben menni is fog. Azért könnyű eldönteni, hogy melyik a kisebb, mert a magasságokat ki tudjuk olvasni a gyökérben. E feltétel nélkül meg kéne határozni a magasságokat is.

7. **[ZH: 2009. június 4.] Mutassa meg, hogyan kell a 2-3 fa BESZÚR eljárását módosítani, ha a fa minden v csúcsában a szokásos dolgokon kívül azt is nyilvántartjuk, hogy hány levél van a v gyökerű részfában!**
Ha nem kell csúcsvágás, akkor a levéltől a gyökérig végig megnöveljük eggyel a tárolt értéket (ez belefér az $O(\log n)$-be). Ha kell, akkor a vágott csúcsokra újra kell állítani az értéket (a levelek fölött ez triviális, feljebb pedig a gyerekekből konstans időben kiolvasható és összeadható), egyébként a $+1$ ugyanúgy terjed felfelé. Ez az eset is belefér $O(\log n)$-be.

8. **[PZH: 2008. május 9.] Vázolja a 2-3 fának (és műveleteinek) egy olyan módosítását, amiben továbbra is van KERES, BESZÚR, TÖRÖL, MIN, MAX művelet, és ezeken kívül van még RANG és K-ADIK művelet is, ahol RANG(x) azt adja vissza, hogy a tárolt elemek között az x a rendezés szerint hányadik elem, a K-ADIK(i) pedig, hogy a rendezés szerint a tárolt elemek közül melyik az i-edik. A módosítás során a felsorolt szokásos műveletek lépésszámának nagyságrendje ne változzon, és mindkét új művelet lépésszáma legyen $O(\log n)$, ahol n a tárolt elemek száma.**
Lényegében ugyanaz, mint a következő feladat.

9. **[Vizsga: 2003. március 31.] Tervezzen adatstruktúrát a következő feltételekkel. Természetes számokat kell tárolni, egy szám többször is szerepelhet. A szükséges műveletek:**
BESZÚR(i): i egy újabb példányát tároljuk
TÖRÖL(i): i egy példányát töröljük
MINDTÖRÖL(i): i összes példányát töröljük
DARAB(i): visszaadja, hogy hány példány van i-ből
ELEM(K): megmondja, a nagyság szerinti rendezésben a K-adik elem értékét.
Az adatstruktúra legyen olyan, hogy ha m-*féle* elemet tárolunk, akkor mindegyik művelet lépésigénye $O(\log m)$.
(Például ha a tárolt elemek $1, 1, 3, 3, 3, 8$, akkor DARAB$(1) = 2$, ELEM$(4) = 3$ és $m = 3$.)

A levelekben nem csak a konkrét értéket tároljuk, hanem egy számlálót is, hogy az adott elemből hányat tárolunk (így kicsit módosul a törlés és beszúrás). Azt is nyilvántartjuk továbbá minden csúcsra, hogy a belőle kiindulú részfában hány elem van (ez ugyanaz, mint a kettővel ezelőtti feladat, csak a többszörös elemkre is kell egy kicsit figyelni). A K-adik elem egy keresés, ahol (a részfaméretek alapján) mindig arra megyünk, ahol még épp nem lépjük túl K-t. A fának m levele lesz, a szokásos műveletek nem változnak, így a műveletek $O(\log m)$-ben menni fognak. (Természetesen célszerű szépen leírni az egyes műveleteket, ez csak egy vázlat. Továbbá p-f fával is meg lehetne oldani a feladatot, bár azzal kicsit macerább.)

10. **[ZH: 2003. március 31.] Egy 2-3 fába egymás után 1000 új elemet illesztettünk be. Mutassa meg, hogy ha ennek során egyszer sem kellett csúcsot szétvágni, akkor a beillesztések sorozata előtt már legalább 2000 elemet tároltunk a fában.**
    Ha nem volt csúcsvágás, akkor mind az 1000 elemet egy 2 gyerekű csúcsba kellett beszúrni. Ilyen (a triviális, 1 elemet tárolós esetet leszámítva) csak vágás során keletkezhet, így mind az 1000 elemet egy eredetileg is létező 2 gyerekű csúcsba kellett beszúrni, vagyis összesen legalább 2000 elem biztos, hogy már a fában volt.

11. **[ZH: 2004. március 29.] Egy kezdetben üres 2-3 fába az $1, 2, \ldots, n$ számokat szúrtuk be ebben a sorrendben. Bizonyítsa be, hogy a keletkezett fában a harmadfokú csúcsok száma $O(\log n)$.**
    Ezt így elég macera lenne bizonyítani, ezért egy ennél erősebb dolgot fogunk. Állítás: az ilyen beszúrások során kizárólag a legjobboldali út mentén vannak harmadfokú csúcsok. Indukcióval $n = 1\ldots$ kevés esetre könnyen látszik. Tfh n-re igaz, most beszúrjuk $n+1$-et. Ez nyilván a legjobboldalra megy, ha ott éppen egy másodfokú csúcs van, akkor a feltételnek megfelelően csak ott hoz létre egy harmadfokút, más nem változik. Ha csúcsot vágunk, akkor másodfokúak keletkeznek, ami nem zavar minket, valamint a feljebbi szinten ugyanúgy a legjobboldalon keletkezik egy beszúrási igény. Ez felfele rekurzívan pedig pont ugyanúgy működik, ahogy azt láttuk.
    Mivel az $O(\log n)$ magasságú fában csak egy gyökér-levél úton lehetnek harmadfokú csúcsok, ezért a számuk is legfeljebb $O(\log n)$.

ZH howto

- Pontos információ (pl. terembeosztás, követelmények) a honlapon.
- Összetűzött, üres lapok, írószer, igazolvány (nem diák), név, neptun, neptun szerinti gyakvez.
- Bármilyen segédeszköz (számológép, puska, stb.) tiltott. A rosszul viselkedőket elküldjük.
- A helyes indoklás a jó eredmény kulcsa! Az indoklás nélküli eredményközlés veszélyes.