Algel VI. gyakorlat

Keresünk, fázunk

2011. március 19.

4. **[ZH: 2004. március 29.] Egy bináris keresőfában csupa különböző egész számot tárolunk. Lehetséges-e, hogy egy $KERES(x)$ hívás során a keresési út mentén a $20, 18, 3, 15, 5, 8, 9$ kulcsokat látjuk ebben a sorrendben? Ha nem lehetséges, indokolja meg miért nem, ha pedig lehetséges, határozza meg az összes olyan x egész számot, amire ez megtörténhet.**
   Lehet, hiszen az $x < 20$, $x < 18$, $x > 3$, $x < 15$, $x > 8$, $x \neq 9$-et kielégítő szám lehet, vagyis $9 < x < 15$ közül bármi jó, valamint a keresőfa tulajdonság sem sérül sehol.

5. **[ZH: 2009. április 24.] Egy bináris fa csúcsai 0 és 9 közötti egész számokkal vannak megcímkézve. Az inorder bejárás során a címkék sorrendje: 9, 3, 1, 0, 4, 2, 7, 6, 8, 5, a postorder bejárásnál pedig 9, 1, 4, 0, 3, x, 7, 5, y, 2. Mi lehet az x és mi az y?**
   2=gyökér (postorder miatt), 7,6,8,5 a jobb részfában van, többi a balban (inorder miatt) (3 pont). Két lehetőségünk van, $x = 6, y = 8$ és $x = 8, y = 6$. Ezeket kipróbálva (az akutális részfa gyökerét meghatározva, mint először) kijön, hogy előbbi lehet (3 pont), utóbbi nem (4 pont) (ezeket végig kell számolni!).

6. (a) **Lehet-e tetszőleges (adott) kulcshalmaz esetén olyan piros-fekete fát építeni, hogy az azonos szinten lévő elemek azonos színűek legyenek?**
   Két csúcs esetén pl. nem lehet, hiszen a gyereknek muszáj pirosnak lennie, míg a testvére (aki levél), fekete.

   (b) **Van-e olyan piros-fekete fa, ami nem így néz ki?**
   Mi az hogy, nagyon is! Mondjuk az előző példában szereplő.

7. **[ZH: 2007. április 27.] Egy piros-fekete fában lehetséges-e, hogy a piros-fekete tulajdonság megsértése nélkül**

   (a) **néhány fekete csúcsot átváltoztathatunk pirosra?**
   Van olyan fa, amiben ez igaz (pl. teljes fekete fa), de van olyan is, ahol nem (pl. 2 db értéket tároló fa).

   (b) **valamelyik (csak egy) fekete csúcsot átváltoztathatjuk pirosra?**
   Biztos, hogy nem, hiszen a gyökeret nem pirosíthatjuk be, más helyen viszont ez aszimmetrikusan megváltoztatná a fekete magasságot.

   (Mást nem változtatunk a fán.)

8. **Egy bináris keresőfa csúcsait egy, a gyökértől egy levélig menő út szerint három osztályba soroljuk: B az úttól balra levő, U az útra eső, J pedig az úttól jobbra levő csúcsok halmazát jelöli. Igaz-e mindig, hogy minden B-beli csúcs kulcsa kisebb tetszőleges U-beli csúcs kulcsánál, és minden U-belis csúcs kulcsa kisebb tetszőleges J-beli csúcs kulcsánál?**
   Nem, ellenpéldát lehet rá adni egyszerűen: a gyökérben mondjuk legyen 1, ennek (egyetlen) fia 3, ennek pedig két fia, 2 és 4. Az út: $\{2, 3, 4\}$, ettől balra van 2, ami kisebb, mint 1.

9. **Adott n pont a síkon, melyek páronként mindkét koordinátájukban különböznek. Bizonyítsuk be, hogy pontosan egy bináris fa létezik, melynek csúcsai az adott n pont, és az első koordináta szerint a keresőfa tulajdonsággal, a második szerint a kupac tulajdonsággal rendelkezik! (A kupac tulajdonságba most nem értjük bele, hogy a fa teljes bináris fa legyen.)**

A gyökér mindenképp y szerint a legkisebb lesz, a kupac tulajdonság miatt. A keresőfa tulajdonság miatt egyértelmű, hogy kik lesznek a bal-, és kik a jobb részfában, de itt megint egyértelmű a kupac tulajdonság miatt, hogy ki lesz az első, és így tovább rekurzívan.

10. **[Vizsga: 2009. május 28.] Adott egy n csúcsú bináris keresőfa. Ennek minden v csúcsára meg akarjuk határozni, hogy a v gyökerű részfában hány darab v-nél kisebb elem van tárolva. Adjon algoritmust, ami ezt a feladatot $O(n)$ lépésben megoldja!**
Keresőfát definiáljuk (1 pont). Minden egyes pontban a bal részfák mérete kell (1 pont). Algoritmus (dinprog): a csúcsokban (v) két számot tartunk nyilván: a bal részfa méretét (b_v), valamint a teljes részfa méretét (f_v). Levelekben $b_v = 0$, $f_v = 1$ (ez nyilván helyes). Lentről felfele töltjük az értékeket, ha v gyerekei i és j, akkor $b_v = f_i$, $f_v = f_i + f_j + 1$ (ha a lentebbi értékek helyesek, akkor induktívan ez is helyes lesz). (algo: 4 pont, indoklás: 2 pont). Minden csúcsot legfeljebb 2-szer vizsgálunk (amikor őt, és amikor a szülőjét töltjük), valamint a csúcsokon megfelelően végigmenni egy bejárással lineáris idő, így a lépsszám $O(n+2n) = O(n)$ (2 pont).

11. **[ZH: 2009. április 24.] Egy piros-feket fában jelölje x és y a gyökér két fiát. Tudjuk, hogy $fm(x) = fm(y)$, de az x csúcs két gyerekének különbözik a fekete magassága. Milyen színű lehet az y csúcs?**
Fekete magasság definíciója (1 pont). x és y azonos színű, mert ha nem lenne az, akkor a gyökérben a fekete magasság a két ágon különböző lenne (3 pont). x gyerekei különböző színűek, mert ha egyformák lennének, akkor a két ágon sérülne x-ben a fekete magasság (1 pont). x-nek tehát van piros gyereke, így ő kötelezően fekete (4 pont). Ezekből következik, hogy y is fekete (1 pont).

12. **Adott egy $n = 2^k - 1$ pontú teljes bináris keresőfa. A fában tárolt elemek egészek az $I = [1, 2^k]$ intervallumból, és egy szám legfeljebb egyszer fordul elő a fában. Utóbbi feltétel szerint pontosan egy olyan i egész szám van 1 és 2^k között, amely nincs a fában. Adjunk egy hatékony módszert i meghatározására!**
A gyökér bal és jobb részfájában pontosan $2^{k-1} - 1$ elem van, a bal oldaliak nagyság szerint a gyökér előttiek, a jobb oldaliak a gyökér utániak. Ha a bal oldalról hiányzik valaki, akkor a gyökér $2^{k-1} + 1$, ha a jobb oldalról, akkor 2^{k-1}. Ezzel visszavezettük a feladatot egy eggyel kisebb magasságúra, tehát a lépésszám a szintszámmal arányos, azaz $O(\log n)$.

13. **Egy fában az x csúcs *súlya* legyen x leszármazottainak száma. Egy bináris fát szigorúan binárisnak mondunk, ha a levelek kivételével minden csúcsnak pontosan 2 fia van. Tegyük fel, hogy egy szigorúan bináris fa minden x csúcsára fennáll, hogy**

$$\frac{1}{2} < \frac{súly(bal(x))}{súly(jobb(x))} < 2.$$

Bizonyítsuk be, hogy ez csakis egy teljes fa lehet, azaz ha k szintje van, akkor a csúcsok száma $2^k - 1$. *(Ez nem kifejezetten keresőfázós feladat, de úgy általában érdekes.)*
Indukcióval szintszám szerint. $l = 1$ szintre biztos igaz. Tfh igaz valamilyen $l-1$-ig, belátjuk l-re is. Az indukciós feltétel szerint a bal és jobb részfája is teljes bináris fa, a bennük tárolt elemek száma $2^{l_b} - 1$ és $2^{l_j} - 1$, ha a bal- és jobb részfa szintszáma l_b és l_j. Tfh $l_b > l_j$, ekkor $\frac{2^{l_b}-1}{2^{l_j}-1} > \frac{2 \cdot 2^{l_j}-1}{2^{l_j}-1} > \frac{2 \cdot 2^{l_j}-2}{2^{l_j}-1} = 2$ (az első egyenlőtlenség azért igaz, mert a bal részfa legalább 1 szinttel magasabb, mint a jobb, a második pedig azért, mert a nevezőből egy nagyobb számot vonunk ki, így a tört értéke csökken), ami ellentmond a feltételnek. Hasonlóan $l_j > l_b$ is lehetetlen, így $l_b = l_j$, pont, amit bizonyítani akartunk.