Algel VI. gyakorlat

Keresőfázunk

2012. március 12.

1. Rendezzük a következő számsorozatot ládarendezéssel, ha tudjuk, hogy 0 és 10 közötti egész számok fordulhatnak csak elő!
   7,8,4,5,5,4,5,0,3,4,7,5

2. Rendezzük a következő láncokat a radix rendezés segítségével: $abc, acb, bca, bbc, acc, bac, baa$!

3. Építsünk a naiv algoritmussal keresőfát a következő elemekből (a rendezés ABC szerint történik): D, B, E, A, C, F, majd töröljük a következő elemeket: F, D!

4. Építsünk piros-fekete fát a következő számokból: 7, 8, 2, 10, 5, 4. Járjuk be a fát pre-, in-, és postorder módon!

5. **[Vizsga: 2007. június 5.]** Adott egy n csúcsú és egy k csúcsú piros-fekete fa. A két fában tárolt összes elemből $O(n + k)$ lépésben készítsen rendezett tömböt.

---

6. **A (növekvően) rendezett $A[1 : n]$ tömb k elemét valaki megváltoztatta. A változtatások helyeit nem ismerjük. Javasoljunk $O(n + k \log k)$ költségű algoritmust az így módosított tömb rendezésére!**
   Menjünk végig a tömbön, és a hibás párokat (amik rosszul állnak) szedjük külön. Ez azért kell, mert a rossz szomszédok közül legalább az egyiket biztos megváltoztattuk (hiba csak változtatásnál lehet), viszont nem tudjuk, melyiket. Így legfeljebb $2k$ elemet vettünk ki, ezeket $O(2k \log 2k) = O(k \log k)$ időben tudjuk rendezni, utána $O(n+k)$ időben összefésülés. A kezdeti végigolvasás $O(n)$, vagyis összességében tényleg $O(n + k \log k)$ lépésben végzünk.

7. **Vázoljunk egy $O(n)$ időigényű algoritmust (az időkorlát bizonyításával együtt) n olyan egész számból álló sorozat rendezésére, melynek elemei az**

   (a) **$\{1, \ldots, 3n\}$ tartományba esnek!**
   Ládarendezés, $3n$ ládával, $O(n + 3n) = O(n)$.

   (b) **$\{1, \ldots, n^7 - 1\}$ tartományba esnek!**
   n-es számrendszerben felírjuk a számokat (ez darabonként 7 osztás, vagyis konstans), utána számjegyenként radix (az n alapú számrendszer miatt legfeljebb 7 jegyűek lettek a számok), azaz $O(7n + 7n) = O(n)$.

8. **[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.

9. **[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!).

10. (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.

11. **[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. legalább 3 elemet tartalmazó teljes fekete fa), de van olyan is, ahol nem (pl. 2 db értéket tároló fa). És mivel a kérdés úgy értelmezendő, hogy „tetszőlegesre igaz-e", ezért az ellenpélda bizonyítja a nemleges választ.

    (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. (Természetesen ide is elég lenne egy ellenpélda.)

    (Mást nem változtatunk a fán.)

12. **[ZH: 2011. április 19.] Adott $2^k - 1$ különböző szám, mindegyik az $\{1, 2, \ldots, n\}$ halmazból, ezekből kell egy $O(k)$ mélységű bináris keresőfát készíteni. Adjon olyan algoritmust, amely ezt $O(n)$ lépésben megcsinálja!**
$2^k - 1$ számot tudunk egy pontosan k szintű teljes bináris keresőfában tárolni, így ilyet fogunk csinálni. Mivel a fa alakja adott, csak azt kell kitalálni, hogy melyik szám melyik csúcsba kerüljön. Viszont azt is tudjuk, hogy egy bináris keresőfát inorder bejárva a benne tárolt elemeket növekvő sorrendben kapjuk meg. Így meg tudjuk tenni, hogy felépítünk egy „biankó" k szintű teljes bináris fát, futtatunk rá egy inorder bejárást, és a növekvően rendezett elemeinkből egy csúcs meglátogatásakor mindig odarakjuk a következőt. A számok különbözősége miatt $2^k - 1 \leq n$, így a fa felépítése $O(2^k - 1) = O(n)$, a számok ládarendezése (minden feltétel adott az alkalmazhatósághoz) $O(2^k - 1 + n) = O(n)$, majd a bejárás $O(2^k - 1) = O(n)$. Ezeket összegezve a lépésszám $O(n)$.

13. **[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).

14. **[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).

15. **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: $\{1, 3, 4\}$, ettől balra van 2, ami nem kisebb, mint 1.

16. **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.

17. **A 4 elemű I abc felett adott két szó: $x = x_1x_2 \dots x_n$ és $y = y_1y_2 \dots y_k$, ahol $1 \leq k \leq n$ és $\forall i, j : x_i, y_j \in I$. Keressük az x szóban az olyan részszavakat, amelyek anagrammái y-nak, azaz az olyan i indexeket, hogy az $x_i, x_{i+1}, \dots, x_{i+k-1}$ betűk megfelelő sorrendbe rakva az y szót adják. Adjunk algoritmust, ami x-ben az összes ilyen i helyet $O(n)$ lépésben meghatározza!**
Csináljunk egy hisztogramot y-ból, vagyis egy 4 elemű tömbbe írjuk be, hogy melyik karakterből hányat tartalmaz (magyarul ládarendezés). Ez $O(k) = O(n)$. Ezután csináljunk egy hasonló tömböt x első k elemére (T_x), ami szintén ugyanennyi idő. Ha a két tömb megegyezik (ezt a 4 elem miatt $O(4) = O(1)$-ben tudjuk ellenőrizni), akkor találtunk egy anagrammát. Ha az ablakot (ami általánosan az x_i-nél kezdődik) eggyel jobbra mozgatjuk, akkor $--T_x[x_i]$ és $++T_x[x_i + k + 1]$ (azaz az elhagyott karaktert kivesszük, a következőt betesszük). A helyesség könnyen látható, lépésszám: mind az n pozícióra (igazából kicsit kevesebbre) két tömbművelet és egy 4 hosszú egyenlőségvizsgálat (azaz összességében konstans), tehát $O(n)$-ben ez a része is megvan.

18. **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 $O(\log n)$ lépésszámú algoritmust 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, amit rekurzívan ugyanígy tudunk tovább kezelni, tehát a lépésszám a szintszámmal arányos, azaz $O(\log n)$.

19. **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.