

Algel VI. gyakorlat

Rendezgetünk, Keresünk, fázunk

2014. március 20.

1. 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!
 2. Vázoljunk egy $O(n)$ időigényű algoritmust (az időkorlát bizonyításával együtt) n olyan egész számból álló sorozat rendezésére, melynek elemei az
 - (a) $\{1, \dots, 3n\}$ tartományba esnek!
 - (b) $\{1, \dots, n^7 - 1\}$ tartományba esnek!
 3. **[Vizsga: 2004. június 3.]** A $2^k - 1$ elemű A tömb elemei mind különbözőek és növekvő sorrendben vannak. Minden elemet egy k hosszú bitsorozat ír le, tehát tekinthetjük úgy, hogy a $0, 1, 2, \dots, 2^k - 1$ számokat tároljuk egy kivételével. A feladat ennek a hiányzó számnak a megkeresése. Ehhez egy lépésben valamelyik elem egy bitjére kérdezhetünk rá: a $BIT(i, j)$ eljárás az $A[i]$ elem j -edik bitjét mondja meg. Adjon olyan algoritmust, amely a BIT eljárás $O(k)$ -szori hívásával megtalálja a hiányzó számot (bitsorozatot).
 4. Adott egy dobozban n különböző méretű anyacsavar, valamint egy másik dobozban a hozzájuk illő apacsavarok. Kizárólag a következő összehasonlítási lehetőségünk van: egy apacsavarhoz hozzápróbálunk egy anyacsavart. A próbának háromféle kimenete lehet: $\text{apa} < \text{anya}$, $\text{apa} = \text{anya}$, $\text{apa} > \text{anya}$, annak megfelelően, hogy az apacsavar külső átmérője hogyan viszonyul az anyacsavar belső átmérőjéhez. Szeretnénk minden anyacsavarhoz megtalálni a megfelelő apacsavart. Adjunk erre a feladatra *átlagosan* $O(n \log n)$ összehasonlítást felhasználó módszert!
-
5. Építsünk a naiv algoritlussal keresőfát a következő elemekből (a rendezés ABC szerint történik): D, B, E, A, C, F , majd töröljük a következő elemeket: F, D !
 6. **[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.
-
7. **[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.
 8. **[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 ?
 9. Egy bináris keresőfa csúcsait egy, a gyökértől egy levélíig 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 -beli csúcs kulcsa kisebb tetszőleges J -beli csúcs kulcsánál?
 10. 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.)
 11. **[Vizsga: 2009. május 28.]** Adott egy n csúcsú bináris keresőfa. Ennek minden v csúcsára meg akarjuk határozni, hogy a v gyökerű részében hány darab v -nél kisebb elem van tárolva. Adjon algoritmust, ami ezt a feladatot $O(n)$ lépésben megoldja!

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!
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.)*