Kazi Sándor
KAZI(AT)CS.BME.HU

Kulcsmanipulációs rendezés, bináris keresőfa

1. Adott egy $n \times n$-es mátrix. Adjunk $O(n^2 \log n)$ összehasonlítást használó algoritmust, amely eldönti, van-e két olyan sor, amelyeknek az első oszlopbeli elemei különböznek, viszont az összes többi oszlopban megegyeznek!
2. Hogyan működik a láda- és a (lexikografikus) radix-rendezés?
3. Rendezzük radix rendezéssel az acedf, adfce, ecdaf, acfde, edcaf, edcfa, caedf elemeket. Hol jelenik meg az a mennyiség, ami miatt lehet ládarendezést alkalmazni.
4. Egy azonos hosszú abc-szerint rendezett (ismétlődő elem nélküli) karaktersorozatokból álló lista kolexikografikusan rendezett, ha bármely két i és j elem esetén i pontosan akkor előzi meg j-t, ha az i lexikografikus ("normál") rendezés szerinti utolsó karaktere megelőzi j lexikografikusan utolsó elemét. (Pl.: abc, abd, acd, bcd, abe, ace, ade, bce, $\dots$) Hogyan rendezhetnénk sorba a feltételnek megfelelően azonos hosszú, abc-szerint rendezett karaktersorozatokat. Mennyi ennek a költsége? Rendezzük eszerint ezeket: cdf, bcf, ace, cef, cde, bcd.
5. 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!
6. Milyen egy bináris keresőfa struktúrája? Miben különbözik a bináris kupactól? Hogyan járja be a fát pre-, az in- és a postorder bejárás?
7. Mely bejárásoknál lehetséges az, hogy a tárolt elemek legnagyobbika megelőzi a legkisebbet?
8. Egy bináris fa inorder bejárása: j, b, k, g, i, a, c, d, f, e, h preorder bejárása: a, b, j, g, k, i, d, c, e, f, h. Rekonstruáljuk a fát!
9. Egy bináris fa preorder bejárása a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, postorder bejárása e, d, c, h, i, g, f, b, l, n, o, m, k, j, a. Egyértelmű-e ebből az inorder bejárás? Ha igen, akkor mi a csúcslista, ha nem, akkor miért nem?
10. 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, indokoljuk meg miért nem, ha pedig lehetséges, határozzuk meg az összes olyan x egész számot, amire ez megtörténhet.
11. Mik a piros-fekete fák? Hogyan definiáljuk? Mit biztosítanak a piros-fekete fák az elemek kereshetőségére vonatkozóan a bináris keresőfákhoz képest?
12. Melyik az a 10 pontú piros-fekete fa, aminek a legnagyobb a magassága?
13. Hány csúcsa van egy piros-fekete fának, ha k elemet tárolunk benne?
14. Milyen az piros-fekete fa, aminek minden csúcsa fekete?
15. Mennyi a tárolható elemek számának minimuma, illetve maximuma egy olyan piros-fekete fában, aminek a fekete magassága 3, illetve ha 4?
16. Adott egy n csúcsú és egy k csúcsú piros-fekete fa. A két fában tárolt összes elemből $O(n + k)$ lépésben szeretnénk készíteni egy rendezett tömböt, tudjuk, hogy meg lehet csinálni. Adjunk megfelelő algoritmust!
17. Szúrjuk be egy bináris keresőfába a 6, 2, 1, 4, 5, 8, 9, 7, 10 elemeket, majd töröljük a 7, 8, 2 elemeket!
18. Lehetséges-e, hogy egy piros-fekete fából a piros-fekete tulajdonság megsértése nélkül pirosról feketére illetve feketéről pirosra átszínezzünk
    (a) néhány csúcsot?
    (b) pontosan egy csúcsot?
    (c) pontosan k csúcsot?