

5. gyakorlat

Rendezések

- Adott az $A[1 : n]$ csupa különböző egész számot növekvő sorrendben tartalmazó tömb. (A tömbben negatív számok is lehetnek!) Adjunk hatékony algoritmust egy olyan i index meghatározására, melyre $A[i] = i$ (feltéve, hogy van ilyen i): igyekezzünk minél kevesebb elem megvizsgálásával megoldani a feladatot!
- Adott az $A[n]$ csupa különböző számot tartalmazó tömb. Tudjuk, hogy $A[1] < A[2]$ és $A[n-1] > A[n]$. Adjunk $O(\log n)$ költségű algoritmust, amely megtalál egy lokális maximumot az A tömbben, ha van ilyen.
- Egy egyszerű operációs rendszerben szeretnénk nem preemptív (egy elkezdett taszk nem szakítható meg) ütemezőt csinálni. Minden taszknak van egy pozitív prioritása, a legkisebb prioritási érték a legfontosabb feladat. Hogyan érdemes tárolnunk a taszkok várakozási sorát, ha az új taszkokat gyorsan akarjuk kiválasztani?
- Adott egy kupac, mely n darab számot tartalmaz. Egy új kupacot szeretnénk építeni az eredeti kupac elemeinek (-1) -szereiből. (Ehhez, ha akarjuk, használhatjuk az eredeti kupacot.) Mutassuk meg, hogy az új kupac elkészítéséhez használt összehasonlítások száma $\Omega(n)!$
- Hogyan működik a beszúrásos, az összefésüléses, a buborék- és a gyorsrendezés eljárás? Milyen lépésszámmal végezhető el egy n elemű lista rendezése ezekkel az algoritmusokkal?
- Rendezzük a következő listát összefésüléses rendezés segítségével: 4, 11, 9, 10, 5, 6, 8, 1, 2, 16.
- Egy csupa különböző egészekből álló sorozat *bitonikus*, ha először nő, utána pedig fogy, vagy fordítva: először fogy, utána nő. Például az (1, 3, 7, 21, 12, 9, 5), (9, 7, 5, 4, 6, 8) és (1, 2, 3, 4, 5) sorozatok bitonikusak. Adjunk $O(n)$ összehasonlítást használó rendező algoritmust n elemű bitonikus sorozatok rendezésére!
- Az $A[1 : n]$ tömbben egy rendezett univerzum n különböző eleme volt, nagyság szerint növekvő sorrendben. Valaki időközben megkeverte a tömb elemeit, de csak annyira, hogy minden egyes elem új helye az eredetitől legfeljebb 5 távolságra esik. Adjunk $O(n)$ idejű algoritmust az eredeti állapot helyreállítására!
- 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: apa < anya, apa = anya, vagy apa > anya; annak megfelelően, hogy az apacsavar külső átmérője hogyan viszonyul az anyacsavar belső átmérőjéhez. Szeretnénk az anyacsavarokhoz megtalálni a megfelelő apacsavarokat. Adjunk erre a feladatra **átlagosan** $O(n \log n)$ összehasonlítást felhasználó módszert!
- 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!
- Hogyan működik a láda- és a (lexikografikus) radix-rendezés?
- Rendezzük radix rendezéssel az acedf, adfce, ecda, acfde, edcaf, edcfa, caedf elemeket.
- Egy azonos hosszú abc-szerint rendezett (ismétlődő elem nélküli) karaktersorozatokból álló lista kolerikografikusan 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, ...*) 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*.
- Vázzunk egy $O(n)$ időigényű algoritmust (az időkorlát bizonyításával együtt) n olyan egész számból álló sorozat rendezésére, melynek elemei az
 - $\{1, \dots, 3n\}$ tartományba esnek!
 - $\{1, \dots, n^7 - 1\}$ tartományba esnek!