

Logikai feladatok, rendezések

1. A helyi földesúrnak van 500 hordó bora, délután 4 van és este 8-kor partit akar tartani. Az a baj, hogy az egyik hordó mérgezett, bárki aki akár egy cseppet is iszik belőle, egy órán belül meghal (azaz rögtön is meghalhat, de legkésőbb egy óra múlva). Van 4 szolgája, akik számára nem képviselnek értéket, így hajlandó feláldozni őket a nemes cél érdekében. Megkóstoltatja velük a borokat tetszőleges kombinációban, majd várja, hogy meghaljanak. Megoldható-e, hogy kiderüljön, melyik a mérgezett bor a parti kezdetéig?
2. Két egyforma strucctojással a kezünkben állunk egy 36 emeletes ház előtt. Szeretnénk megtudni, hogy mi az a legkisebb emeletszám, ahányadikról egy strucctojást leejtve az már összetörik (lehet, hogy már az elsőről leejtve is végük, lehet, hogy a 36 emeletes zuhanást is kibírják). Legalább hány ejtési kísérletre van szükségünk? (Tudományos kíváncsiságunk erősebb, mint a birtoklási ösztönünk, ezért a kísérletek végén nem szükséges, hogy ép strucctojás legyen a birtokunkban.)
3. 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!
4. 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.
5. 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?
6. Adott egy kupac, mely n darab számot tartalmaz. Egy új kupacot szeretnénk építeni az eredeti kupac elemeinek (-1) -szereseiből. (Ehhez, ha akarjuk, használhatjuk az eredeti kupacot.) Mutassuk meg, hogy az új kupac elkészítéséhez használt összehasonlítások száma $\Omega(n)$!
7. Hogyan működik a beszúrásos, az összefésüléses, a buborék- és a gyorsrendezés eljárás? Milyen lépésszámmal végezhető el egy n elemű lista rendezése ezekkel az algoritmusokkal?
8. Rendezzük a következő listát összefésüléses rendezés segítségével: 4, 11, 9, 10, 5, 6, 8, 1, 2, 16.
9. 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!
10. 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!
11. Legyen adott egy rendezett univerzum $2n$ különböző eleméből álló S halmaz. Szeretnénk az S elemeit egy $A[1 : 2n]$ tömbbe elhelyezni úgy, hogy

$$A[1] < A[2] > A[3] < A[4] > \dots < A[2n-2] > A[2n-1] < A[2n]$$

teljesüljön. Adjunk meg egy $O(n)$ összehasonlítást használó algoritmust erre a feladatra!

12. 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: $\text{apa} < \text{anya}$, $\text{apa} = \text{anya}$, vagy $\text{apa} > \text{anya}$; 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!
13. 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!

14. Hogyan működik a láda- és a (lexikografikus) radix-rende­zés?
15. Rendezzük radix rendezéssel az $acedf$, $adfce$, $ecdaf$, $acfde$, $edcaf$, $edcfa$, $caedf$ elemeket.
16. Egy azonos hosszú abc-szerint rendezett (ismétlődő elem nélküli) karaktersorozatokból álló lista kolesikografikusan 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 .
17. 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!