Algel VI. gyakorlat

Még mindig keresünk és rendezünk

2009. március 17/19.

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

2. Egy n elemű sorozat csupa 0-ból és 1-esből áll. Rendezzük a sorozatot $n-1$ összehasonlítással!

3. 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, apa>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!

4. Tegyük fel, hogy adott a $\Sigma = \{a, b, c, d\}$ abc feletti szavaknak egy S halmaza. Tudjuk, hogy a szavak összhossza n. Javasoljunk egy $O(n)$ idejű módszert az S elemeinek (lexikografikus) sorbarendezésére!

5. 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!
   (b) $\{1, \ldots n^7 - 1\}$ tartományba esnek!

6. A 4 elemű I abc felett adott két szó: $x = x_1 x_2 \ldots x_n$ és $y = y_1 y_2 \ldots y_k$, 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 $x_i, x_{i+1}, \ldots, x_{i+k-1}$ betűk megfelelő sorrendbe rakva az y szót adják. Adjunk algoritmust, ami x-ben az összes ilyen i helyet $O(n)$ lépésben meghatározza!

7. **[ZH: 2004. március 29.]** Az $A[1 \ldots n]$ tömbben egész számokat tárolunk, ugyanaz a szám többször is szerepelhet. Határozzuk meg $O(n \log n)$ lépésben az összes olyan számot, amelyik egynél többször fordul elő a tömbben.

8. **[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 $0, 1, 2, \ldots, 2^k - 1$ 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).

9. **[ZH: 2002. április 8.]** Adottak a $c_1, c_2, \ldots, c_n$ különböző egész számok. Ezeket szeretnénk nagyság szerint rendezni növekvő, vagy csökkenő sorrendbe úgy, hogy a szokásos összehasonlítás helyett, most a következő kérdéseket lehet feltenni: *Három kiválasztott elem közül melyik a középső?* Bizonyítsuk be, hogy a leghatékonyabb algoritmus $\Theta(n \log_2 n)$ összehasonlítást használ!