A számítástudomány alapjai

3. gyakorlat, 2013. szeptember 23 illetve 26.

Lineáris és bináris keresés. Rendezések: beszúrásos, összefésüléses, kiválasztásos, buborék-, és gyorsrendezés

1. Az $A[1:n]$ tömbben levő elemekről tudjuk, hogy $A[1] \neq A[n]$. Adjon $c \cdot \log n$ összehasonlítást használó algoritmust, amely talál egy olyan i indexet, hogy $A[i] \neq A[i+1]$! (ZH: 2004. április 8.)
2. Rendezzük az $7, 3, 12, 1, 5, 4$ tömböt (a) beszúrásos rendezéssel, (b) összefésüléses rendezéssel, (c) buborékrendezéssel, (d) gyorsrendezéssel, (e) minimumkiválasztásos rendezéssel!
3. 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 $c \cdot n$ összehasonlítást használó rendező algoritmust n elemű bitonikus sorozatok rendezésére!
4. Négy elem rendezéséhez legalább hány összehasonlítás kell?
5. Adott egy n hosszú, egész számokból álló sorozat, amelynek elemei az $\{1, 2, \ldots 3n\}$ tartományból kerülnek ki. Adjunk $c \cdot n$ időigényű algoritmust a sorozat rendezésére!
6. 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 $c \cdot k$-szori hívásával megtalálja a hiányzó számot (bitsorozatot). (Vizsga: 2004. június 3.)
7. A $\begin{array}{|c|c|c|c|c|c|c|c|}\hline 6 & 4 & 8 & 3 & 7 & 2 & 5 & 1 \\ \hline\end{array}$ tömb rendezése során (a rendező algoritmus néhány lépése után) a következő közbülső állapot jött létre: $\begin{array}{|c|c|c|c|c|c|c|c|}\hline 4 & 6 & 3 & 8 & 7 & 2 & 5 & 1 \\ \hline\end{array}$ Az alább felsorolt, az előadáson tanult módszerek közül mely(ek) alkalmazásakor fordulhatott ez elő?
   a) Beszúrásos rendezés,
   b) Buborékrendezés,
   c) Összefésüléses rendezés,
   d) Gyorsrendezés?
   e) Minimumkiválasztásos rendezés?
8. A valós számokból álló $a_1, \ldots, a_n$ sorozat olyan, hogy az $a_1^2, a_2^2, \ldots, a_n^2$ sorozat egy darabig nő, utána csökken. Adjon $c \cdot n$ összehasonlítást használó algoritmust, ami rendezi az $a_1, \ldots, a_n$ sorozatot. (ZH 2007. ápr. 27.)
9. Legyen adott egy egészekből álló $A[1:n]$ tömb valamint egy b egész szám. Szeretnénk hatékonyan eldönteni, hogy van-e két olyan $i, j \in \{1, \ldots, n\}$ index, melyekre $A[i] + A[j] = b$. Oldjuk meg ezt a feladatot $c \cdot (n \log n)$ időben!
10. Az $A[1 \ldots n]$ tömbben egész számokat tárolunk, ugyanaz a szám többször is szerepelhet. Határozzuk meg $c \cdot (n \log n)$ lépésben az összes olyan számot, amelyik egynél többször fordul elő a tömbben. (ZH 2004. márc. 29.)
11. Egy n elemű sorozat csupa 0-ból és 1-esből áll. Rendezzük a sorozatot $n - 1$ összehasonlítással!
12. Adott a számegyenesen n intervallum, $[a_1, b_1], \ldots, [a_n, b_n]$. Azt akarjuk tudni, hogy együtt milyen hosszú részt fednek le a számegyenesből (azaz, hogy mennyi az $\cup_{i=1}^{n}[a_i, b_i]$ összhossza). Adjon $c \cdot (n \log n)$ lépéses algoritmust ennek a hossznak a meghatározására! (Vizsga: 2009. június 11.)
13. Az $A[1:n]$ tömbben egy rendezett univerzum n különböző eleme volt, eredetileg 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 $c \cdot n$ idejű algoritmust az eredeti állapot helyreállítására!

14. Adott egy egész számokat tartalmazó $A[1..n]$ tömb, amelyben legfeljebb n elempár áll inverzióban egymással (két elem akkor áll inverzióban, ha a nagyobb megelőzi a kisebbet). Igaz-e, hogy a buborékrendezés rendezi az A tömböt
a) legfeljebb n összehasonlítással?
b) legfeljebb n cserével?

15. Az $A[1 \ldots n]$ tömbben egész számokat tárolunk, ugyanaz a szám többször is szerepelhet. Határozzuk meg $c \cdot (n \log n)$ lépésben a leggyakoribb számokat, vagyis azokat, amelyeknél többször semelyik másik szám sem fordul elő a tömbben.