SzA III. gyakorlat, 2013. szeptember 24/26.
Rendezettek vagyunk és jól keresünk

Drótos Márton
drotos@cs.bme.hu

1. Rendezzük a következő listát buborék, kiválasztásos, beszúrásos, összefésüléses és gyorsrendezés segítségével: $[4, 11, 9, 10, 5, 6, 8, 1, 2, 16]$.

2. Rendezzük a következő listát ládarendezéssel, ha tudjuk, hogy csak 0 és 10 közötti egész számok szerepelhetnek benne: $[6, 4, 3, 8, 6, 3, 3, 5, 2]$

3. Legalább hány összehasonlítás kell ahhoz, hogy egy n elemű tömbből egy olyan tagot találjunk meg, ami a tömb 10 legkisebb eleme közé tartozik?

4. Az A tömbben n különböző számot tárolunk. Tudjuk, hogy $A[1] > A[2]$ és $A[n-1] < A[n]$. Adjunk algoritmust, mely $c \cdot \log n$ összehasonlítással megtalál a tömbben egy lokális minimumot (ha van), azaz egy olyan $1 \leq i \leq n$ indexet, hogy $A[i]$ tömbbeli szomszédai nagyobbak, mint $A[i]$.

5. Adottak a $p_0 = (0, 0), p_1 = (x_1, y_1), \ldots p_n = (x_n, y_n), p_{n+1} = (100, 0)$ pontok a síkban ($n \geq 1$) úgy, hogy $1 \leq i \leq n$ esetén x_i és y_i racionális számok, $0 < x_i < 100$, és semelyik három pont nem esik egy egyenesbe. Egyenes szakaszokkal akarjuk ezeket a pontokat valamilyen sorrendben összekötni úgy, hogy egy $n + 2$ csúcsú zárt töröttvonalat kapjunk, amiben a behúzott szakaszok nem metszik egymást. Adjunk egy legfeljebb $c \cdot n \log n$ lépést használó algoritmust annak meghatározására, melyik pontot melyikkel kössük össze!

6. Rendezzük a következő számokat kupacos rendezéssel: 10, 9, 6, 5, 2, 3 (keddi csoportnak következő gyakorlaton)

---

7. A $[6, 4, 8, 3, 7, 2, 5, 1]$ tömb rendezése során (a rendező algoritmus néhány lépése után) a következő közbülső állapot jött létre: $[4, 6, 3, 8, 7, 2, 5, 1]$. Az alább felsorolt módszerek közül mely(ek) alkalmazásakor fordulhatott elő?

   (a) beszúrásos rendezés
   (b) buborékrendezés
   (c) összefésüléses rendezés
   (d) gyorsrendezés

8. Szeretnénk n db SZA-hallgató ZH-ereményeit (csak az összpontszámot) növekvő sorrendben felsorolni. Adjunk erre $c \cdot n$ lépést felhasználó algoritmust!

9. Adjunk $c \cdot n$ lépésszámú algoritmust n olyan egész számból álló sorozat rendezésére, melynek elemei az $\{1, \ldots, 3n\}$ tartományba esnek!

10. **[pótpótZH, 2010. ősz]** A valós számokból álló $a_1, \ldots, a_n$ sorozat olyan, hogy az $a_1^3, a_2^3, \ldots, a_n^3$ sorozat egy darabig nő, utána csökken. Adjunk konstansszor n összehasonlítást használó algoritmust, ami rendezi az $a_1, \ldots, a_n$ sorozatot.

11. 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!

12. **[ZH, 2012. október 11.]** Adott $n + 2$ rendezett tömb, méreteik rendre $1, 1, 2, 4, 8, \ldots, 2^n$. Adjunk olyan eljárást, ami legfeljebb 2^{n+2} összehasonlítással rendezi a tömbökben tárolt rekordokat.

13. Az $A[1 : n]$ tömbben levő elemekről tudjuk, hogy $A[1] \neq A[n]$. Adjunk $c \log n$ összehasonlítást használó algoritmust, amely talál egy olyan i indexet, hogy $A[i] \neq A[i + 1]$!

14. 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. Adjunk olyan algoritmust, amely a BIT eljárás $c \cdot k$-szori hívásával megtalálja a hiányzó számot (bitsorozatot)!

15. ✎ Adott a síkon n pont, melyek koordinátái $(a_1, b_1) \ldots (a_n, b_n)$. Olyan $P = (x, y)$ pontot keresünk a síkon, amire az alábbi összeg minimális.

$$\sum_{i=1}^{n} (|a_i - x| + |b_i - y|)$$

Adjunk algoritmust, ami $c \cdot n \log n$ lépésben meghatároz egy ilyen P pontot!

16. ✎ Adottak a sík egész koordinátájú $P_1 = (x_1, y_1), \ldots, P_n = (x_n, y_n)$ koordinátájú pontjai. Javasoljunk egy legfeljebb $c \cdot n$ lépésszámú módszert olyan $P_i \neq P_j$ pontok kiválasztására, amelyeken átmenő egyenes által meghatározott félsíkok közül az egyik tartalmazza az összes pontot!

---

17. **[ZH, 2012. október 11.]** Kupac-e az $\begin{array}{|c|c|c|c|c|c|c|}\hline 1 & 4 & 2 & 8 & 3 & 7 & 5 \\ \hline\end{array}$ tömb?

18. A 10 elemű A tömb első 8 elemére legyen $A[i] = 2i (1 \leq i \leq 8)$, és tekintsük ezt, mint egy 8 elemű kupacot. Rajzoljuk le az ehhez tartozó fát! Hajtsuk végre rajta a BESZÚR(3), BESZÚR(1), MINTÖR műveletsort!

19. **[pótpótZH, 2012. december 12.]** Hányféleképpen lehet egy 5 méretű tömbbe úgy beleírni az $1, 2, 3, 4, 5$ rekordokat, hogy kupacot kapjunk?

20. Egy orvosi rendelőben a regisztrációnál kell bejelentkezni, ahol az ott dolgozók eldöntik, hogy a beteg az épp rendelő két orvos közül A-hoz vagy B-hez kell kerüljön, vagy bármelyikükhöz kerülhet. Ezen kívül, a beutaló ismeretében, a beteghez egy, a sürgősséget kifejező, számot is rendelnek. Amikor valamelyik orvos végzett egy beteggel, akkor azon betegek közül, akiket nem csak a másik orvos láthat el behívja a legnagyobb sürgősségi számút. Tegyük fel, hogy a kiosztott sürgősségi számok egymástól különbözőek. Írjunk le egy olyan adatszerkezetet, ami abban az esetben ha n beteg várakozik, akkor a regisztráción az új beteg beillesztését, illetve az orvosoknak a következő beteg kiválasztását $c \log n$ lépésben lehetővé teszi!

21. Adott egy n elemet tartalmazó kupac és egy k kulcs. Keressük meg a kupac k-nál kisebb elemeit! Ha m ilyen elem van, akkor az algoritmus $c \cdot m$ elemi lépést használhat.

22. ✎ Egy kupacba beraktunk egy új x elemet, majd végrehajtottunk egy MINTÖR műveletet. Mikor fordul elő, hogy végül az eredeti kupacot kapjuk vissza?

23. ✎ Igazoljuk, hogy egy n elemből álló kupac felépítése legalább $c \cdot n$ összehasonlítást igényel!