

# Algel V. gyakorlat

## Rendezzük sorainkat

2012. március 7.

1. Legalább hány összehasonlítás kell ahhoz, hogy egy  $n$  elemű tömbből egy olyan tagot találjunk meg, ami a tömb 10 legkisebb eleme közé tartozik?
2. [ZH: 2004. április 8.] Az  $A[1 : n]$  tömbben levő elemekről tudjuk, hogy  $A[1] \neq A[n]$ . Adjon  $O(\log n)$  összehasonlítást használó algoritmust, amely talál egy olyan  $i$  indexet, hogy  $A[i] \neq A[i + 1]$ !
3. Rendezzük a következő listát buborék-, beszúrásos- és összefésüléses rendezéssel!  
4,11,9,10,5,6,8,1,2,16
4. [ZH: 2004. március 29.] Az  $A[1 \dots n]$  tömbben egész számokat tárolunk, ugyanaz a szám többször is szerepelhet. Határozzuk meg  $O(n \log n)$  lépésben az összes olyan számot, amelyik egyénél többször fordul elő a tömbben.
5. [ZH: 2009. április 24.] Adottak a  $p_0 = (0, 0), p_1 = (x_1, y_1), \dots, p_n = (x_n, y_n), p_{n+1} = (100, 0)$  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. Adjon egy  $O(n \log n)$  lépésszámú algoritmust annak meghatározására, melyik pontot melyikkel kössük össze!

---

6. [ZH: 2007. április 27.] A valós számokból álló  $a_1^2 \dots a_n^2$  sorozat egy darabig nő, utána csökken. Adjon  $O(n)$  összehasonlítást használó algoritmust, ami rendezi az  $a_1, \dots, a_n$  sorozat!
7. 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  $O(n)$  futásidejű algoritmust az eredeti állapot helyreállítására!
8. [Vizsga: 2007. június 19.] Adott a síkon  $n$  pont, melyek koordinátái  $(a_1, b_1) \dots (a_n, b_n)$ . Olyan  $P = (x, y)$  pontot keresünk a síkon, amire az alábbi összeg minimális.

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

Adjon algoritmust, ami  $O(n \log n)$  lépésben meghatároz egy ilyen  $P$  pontot.

9. Legyen adott egy egészekből álló  $A[1 : n]$  tömb valamint egy  $b$  egész szám. Szeretnénk hatékonyan eldönteni, hogy van-e két olyan  $i, j \in \{1, \dots, n\}$  index, melyekre  $A[i] + A[j] = b$ . Oldjuk meg ezt a feladatot  $O(n \log n)$  időben!
10. [Vizsga: 2009. június 11.] Adott a számegyenesen  $n$  intervallum,  $[a_1, b_1], \dots, [a_n, b_n]$ . 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  $O(n \log n)$  lépéses algoritmust ennek a hosszának a meghatározására!
11. Adottak a sík egész koordinátájú  $P_1 = (x_1, y_1), \dots, P_n = (x_n, y_n)$  koordinátájú pontjai. Javasoljunk  $O(n)$  költségű módszert olyan  $P_i \neq P_j$  pontok kiválasztására, amelyekén átmenő egyenes által meghatározott félsíkok közül az egyik tartalmazza az összes pontot!

12. A (növekvően) rendezett  $A[1 : n]$  tömb  $k$  elemét valaki megváltoztatta. A változtatások helyeit nem ismerjük. Javasoljunk  $O(n + k \log k)$  költségű algoritmust az így módosított tömb rendezésére!
13. **[Vizsga: 2004. június 10.]** Az  $n$  méretű (nem feltétlenül rendezett)  $A$  tömb elemei különböző pozitív egész számok. Adjon algoritmust, amely meghatároz egy  $1 \leq k \leq n$  számot és kiválaszt  $k$  különböző elemet az  $A$  tömbből úgy, hogy a kiválasztott elemek összege nem több, mint  $k^3$ . Ha nincs ilyen  $k$ , akkor az algoritmus jelezze ezt a tényt! Az algoritmus lépésszáma legyen  $O(n \log n)$ ! (Két szám összehasonlítása, összedáása vagy szorzása egy lépésnek számít.)
14. **[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  $0, 1, 2, \dots, 2^k - 1$  számokat tároljuk egy kivétellel. 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).
15. 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:  $\text{apa} < \text{anya}$ ,  $\text{apa} = \text{anya}$ ,  $\text{apa} > \text{anya}$ , 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!
16. A 4 elemű  $I$  abc felett adott két szó:  $x = x_1 x_2 \dots x_n$  és  $y = y_1 y_2 \dots y_k$ , ahol  $1 \leq k \leq n$  és  $\forall i, j : x_i, y_j \in I$ . Keressük az  $x$  szóban az olyan részzavakat, amelyek anagrammái  $y$ -nak, azaz az olyan  $i$  indexeket, hogy az  $x_i, x_{i+1}, \dots, x_{i+k-1}$  betűk megfelelő sorrendbe rakva az  $y$  szót adják. Adjunk algoritmust, ami  $x$ -ben az összes ilyen  $i$  helyet  $O(n)$  lépésben meghatározza!
17. **[ZH: 2002. április 8.]** Adottak a  $c_1, c_2, \dots, c_n$  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!