# Keresés, rendezés

Algoritmuselmélet

3. gyakorlat

2025.

## Keresések.

**Lineáris keresés.**

Egy $n$-méretű tömbben a lineáris keresés lépésszáma $O(n)$.

**Bináris keresés.**

Egy $n$-méretű rendezett tömbben a bináris keresés lépésszáma $O(\log n)$.

## Rendezések.

Adott $n$ elem.

**Buborékrendezés.**

A buborékrendezés lépésszáma $O(n^2)$.

**Beszúrásos rendezés.**

A beszúrásos rendezés lépésszáma $O(n^2)$.

(Láncolt lista esetén az összehasonlítások száma $O(n^2)$, a mozgatásoké pedig $O(n)$. Tömb esetén az összehasonlítások száma bináris kereséssel $O(n \log n)$, a mozgatásoké pedig $O(n^2)$.)

**Összefésüléses rendezés.**

Az összefésüléses rendezés lépésszáma $O(n \log n)$.

(Egy $k$- és egy $l$-hosszú rendezett lista összefésülése $O(k + l)$ lépés.)

**Gyorsrendezés.**

A gyorsrendezés átlagos lépésszáma $O(n \log n)$, de legrosszabb esetben a lépésszáma $O(n^2)$.

1. Rendezzük az 5, 2, 3, 4, 10, 7, 1, 8 tömböt
   (a) beszúrásos,
   (b) összefésüléses,
   (c) gyorsrendezéssel.
2. Dr. Watson azzal állít be Sherlock Holmes-hoz, hogy olyan összehasonlítás-alapú rendezési algoritmust talált, ami úgy rendez akármekkora tömböt, hogy minden egyes tömbbeli szám legfeljebb 2025-ször vesz részt összehasonlításban. Mivel indokolhatja Sherlock Holmes, hogy Watson téved?
3. 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 $A[i] \neq A[i + 1]$.
4. Egy tömbben $n$ elemet tárolunk.
   (a) Adjunk olyan eljárást, ami $O(n \log n)$ összehasonlítást használ annak eldöntésére, hogy az $n$ elem között található-e kettő, amiknek az összege egy előre meghatározott $b$ szám.
   (b) Adjunk olyan eljárást, ami $O(n^2 \log n)$ összehasonlítást használ annak eldöntésére, hogy az $n$ elem között található-e három, amiknek az összege egy előre meghatározott $b$ szám.

5. Az $n$-elemű $A$ tömb különböző egész számokat tartalmaz. Szeretnénk eldönteni, hogy van-e a tömbben 100 olyan elem, melyekre igaz, hogy bármely kettő különbsége legfeljebb 2012. Adjunk algoritmust, amely $O(n \log n)$ lépésben eldönti ezt a kérdést.

6. Adott $n$ szám, $s_1, s_2, \ldots, s_n$, valamint egy $T$ érték. Hogyan lehet $O(n \log n)$ összehasonlítással olyan $1 \leq i \neq j \leq n$ indexeket találni, hogy $s_i + s_j \geq T$ teljesüljön, és az $|s_j - s_i|$ érték minimális legyen?

7. Adott két tömb, mindegyikben $n$ darab különböző egész számot tárolunk.
   (a) Adjunk $O(n \log n)$ lépésszámú algoritmust a két tömb legkisebb közös elemének megtalálására.
   (b) Adjunk $O(n \log n)$ lépésszámú algoritmust, ami eldönti, hogy van-e legalább $n/2$ olyan elem, ami mindkét tömbben benne van.

8. Adottak $c_1, c_2, \ldots, c_n$ különböző egész számok. Ezeket szeretnénk nagyság szerint rendezni növekvő, vagy csökkenő sorrendbe úgy, hogy a szokásos összehasonlítás helyett most azt a kérdést lehet feltenni 1 lépésben, hogy három kiválasztott elem közül melyik esik a rendezés szerint a másik kettő közé. (Például, ha a $c_i = 3$, $c_j = 1$, $c_k = 7$ elemekről kérdezzük meg ezt, akkor a $c_i$ elemet kapjuk válaszként.) Adjunk $O(n \log n)$ kérdést használó algoritmust, aminek kimenete a csökkenő vagy növekvő sorrendben lévő számok. (Azt nem kell az algoritmusnak meghatároznia, hogy csökkenő vagy növekvő-e a sorozat.)

9. 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 kimenetele lehet: apa < anya, apa = anya, apa > 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.

10. Egy csupa különböző egészekből álló sorozat bitonikus, ha először nő, utána pedig fogy, vagy fordítva: először fogy, utána nő. Adjunk $O(n)$ összehasonlítást használó rendező algoritmust $n$-elemű bitonikus sorozatok rendezésére.

11. Adott egy $A[1:n]$ tömb csupa különböző egész számmal és szeretnénk az elemeiből egy olyan $B[1:n]$ tömböt csinálni, amiben az értékek fel-le váltakoznak, azaz $B[1] < B[2] > B[3] < B[4] > \ldots$
   (a) Adjunk erre a feladatra $O(n \log n)$ lépésszámú algoritmust.
   (b) Adjunk erre a feladatra $O(n)$ lépésszámú algoritmust.

12. Az $n$-méretű (nem feltétlenül rendezett) $A$ tömb elemei különböző pozitív számok. Adjunk $O(n \log n)$ lépésszámú 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 pedig nincs ilyen $k$, akkor az algoritmus jelzi ezt a tényt.

13. Adott a síkon $n$ pont, melyek koordinátái $(a_1, b_1), \ldots, (a_n, b_n)$. Olyan $P(x, y)$ pontot keresünk a síkon, amelyre a $\sum_{i=1}^{n} \left(|a_i - x| + |b_i - y|\right)$ összeg minimális. Adjunk olyan algoritmust, amely $O(n \log n)$ lépésben meghatároz egy ilyen $P$ pontot.

14. 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 $\bigcup_{i=1}^{n} [a_i, b_i]$ összhossza). Adjunk $O(n \log n)$ lépésszámú algoritmust ennek a hossznak a meghatározására.

15. A (növekvően) rendezett $A[1:n]$ tömb $k$ darab elemét valaki megváltoztatta. A változtatások helyeit nem ismerjük. Javasoljunk $O(n + k \log k)$ költségű algoritmust az így módosított tömb rendezésére.