

1. Rendezzük a 6, 3, 1, 4 tömböt (a) buborék-, (b) beszúrásos, (c) összefésüléses, (d) gyorsrendezéssel. Hány összehasonlítás történik az algoritmusok futása során?
2. Adott egy n darab különböző egész számot tartalmazó, rendezett T tömb. Adjunk $O(\log_2 n)$ lépésszámú algoritmust egy olyan i index meghatározására, melyre $T[i] = i$, feltéve, hogy van ilyen index.
3. Adott egy n darab számot tartalmazó T tömb. Határozzuk meg $O(n \log_2 n)$ lépésben az összes olyan számot, amelyik egynél többször fordul elő a tömbben. (zh, 2004.)

4. A $T[1 : n]$ tömbben lévő elemekről tudjuk, hogy $T[1] \neq T[n]$. Adjunk $O(\log_2 n)$ lépésszámú algoritmust, ami talál egy olyan i indexet, amire $T[i] \neq T[i + 1]$. (pzh, 2004.)
5. Adott egy n darab különböző egész számot tartalmazó T tömb. Adjunk olyan $O(n \log_2 n)$ lépésszámú algoritmust, ami eldönti, hogy van-e a tömbben 100 olyan elem, melyekre igaz, hogy bármely kettő különbsége legfeljebb 2026.
6. Adott egy n darab számot tartalmazó T tömb és egy b szám. Adjunk olyan $O(n \log_2 n)$ lépésszámú algoritmust annak eldöntésére, hogy a T tömbben található-e két olyan elem, amiknek az összege éppen b .
7. 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 $O(n \log_2 n)$ összehasonlítást felhasználó módszert.
8. Dr. Watson azzal állít be Sherlock Holmes-hoz, hogy olyan összehasonlítás-alapú rendezési algoritmust talált, ami úgy rendez akármeekkora tömböt, hogy minden egyes tömbbeli szám legfeljebb 2026-szor vesz részt összehasonlításban. Mivel indokolhatja Sherlock Holmes, hogy Watson téved?

9. Egy 7-elemű, rendezett T tömbben 0 és 9 közötti különböző egész számokat tárolunk. Egy x szám bináris keresése során x -et sorban a 6, 2, 4 számokkal hasonlítottuk össze. Adjuk meg az összes T tömböt és x egész számot, amire ez megtörténhetett.
10. A T tömbben n különböző számot tárolunk. Tudjuk, hogy $T[1] > T[2]$ és $T[n - 1] < T[n]$. Adjunk olyan $O(\log_2 n)$ lépésszámú algoritmust, ami megtalál a tömbben egy lokális minimumot, azaz egy olyan i indexet, amire $T[i - 1] > T[i] < T[i + 1]$. (zh, 2010.)
11. Adott két tömb, mindkettőben n darab különböző egész számot tárolunk. Adjunk $O(n \log_2 n)$ lépésszámú algoritmust a két tömb legkisebb közös elemének megtalálására.
12. Adottak az a_1, a_2, \dots, a_n egész számok. Azt akarjuk eldönteni, hogy el lehet-e hagyni közülük legfeljebb tízet úgy, hogy a megmaradt számok összege egy adott b szám alatt maradjon. Adjunk erre a feladatra egy $O(n)$ lépésszámú algoritmust. (vizsga2, 2017.)
13. Egy csupa különböző egészekből álló sorozat bitonikus, ha először nő, utána pedig csökken, vagy fordítva, azaz először csökken, utána nő. Adjunk $O(n)$ lépésszámú algoritmust n -elemű bitonikus sorozatok rendezésére.
14. Adottak 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 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 $O(n \log_2 n)$ 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.) (zh1, 2024.)
15. Egy n számot tartalmazó, rendezett T tömb k darab elemét valaki megváltoztatta. A változtatások helyeit nem ismerjük. Adjunk $O(n + k \log_2 k)$ lépésszámú algoritmust az így módosított tömb rendezésére.