Keresés, rendezés

ALGORITMUSELMÉLET

2. gyakorlat

2024.

Keresések.

Lineáris keresés.

A lineáris keresés lépésszáma $O(n)$.

Bináris keresés.

A bináris keresés lépésszáma $O(\log n)$.

Rendezések.

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 $O(n^2)$.

Ládarendezés.

Ha a rendezendő elemek egy m-elemű halmazból kerülnek ki (pl. az $\{1, 2, \ldots, m\}$ halmazból), akkor a ládarendezés lépésszáma $O(m+n)$.

Radixrendezés.

Ha a rendezendő elemek k komponensből állnak és tetszőleges $i \in \{1, 2, \ldots, k\}$ esetén az i-edik komponensek egy m_i-elemű halmazból kerülnek ki, akkor a radixrendezés lépésszáma $O\left(kn + \sum_{i=1}^{k} m_i\right)$.

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 2024-szer 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. 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.

9. 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.

10. 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.

11. 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.

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

13. 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.

14. Adjunk minél kevesebb összehasonlítást használó algoritmust, ami n elem közül megtalálja a legkisebbet és a legnagyobbat is.