Rendezések

Algoritmuselmélet

7. gyakorlat

2023.

1. (a) Építsünk kupacot a 31, 6, 50, 7, 2, 51 tömbből.
   (b) Szúrjuk be az így kapott tömbbe az 1, majd ezután az 5 számot.
   (c) Hajtsunk végre két egymást követő MINTÖR-t az így kapott kupacon.
2. Rendezzük a 11, 3, 27, 2, 5, 1, 4, 8 tömböt kupacos rendezéssel.
3. Rendezzük a *cab*, *abb*, *cca*, *bcb*, *cbc*, *aac*, *bac* elemeket radix rendezéssel.
4. Vázoljunk egy $O(n)$ időigényű algoritmust n olyan egész számból álló sorozat rendezésére, melynek elemei az
   (a) $\{1, \ldots, 3n\}$ tartományba esnek;
   (b) $\{1, \ldots, n^7 - 1\}$ tartományba esnek.
5. Éllistájával adott egy egyszerű, irányítatlan gráf. Adjunk $O(n + m)$ lépésszámú algoritmust, ami meghatározza a leggyakoribb fokszámot a gráfban (n és m szokás szerint a gráf csúcs-, illetve élszámát jelölik).
6. Adjunk hatékony algoritmust egy kupac tizedik legkisebb elemének megtalálására.
7. Igazoljuk, hogy egy n elemből álló kupac felépítése $\Omega(n)$ összehasonlítást igényel.
8. Bizonyítsuk be, hogy egy n elemből álló kupac legnagyobb elemének megkereséséhez $\Omega(n)$ összehasonlítás szükséges.
9. Egy n-elemű kupac egyik elemét megváltoztattuk. Feltéve, hogy a változtatás helye ismert, adjunk $O(\log n)$ lépésszámú algoritmust a kupac-tulajdonság helyreállítására.
10. 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 2023-szor vesz részt összehasonlításban. Mivel indokolhatja Sherlock Holmes, hogy Watson téved?
11. 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 $A[i] \neq A[i + 1]$.
12. 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.
13. 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.

14. Adott n szám, s_1, s_2, …, 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?

15. Adott két tömb, mindegyikben n darab különböző egész számot tárolunk. Adjunk $O(n \log n)$ lépésszámú algoritmust a két tömb legkisebb közös elemének megtalálására.

16. 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.

17. 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.

18. 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.

19. Adott egy $n \times n$-es mátrix. Adjunk $O(n^2 \log n)$ összehasonlítást használó algoritmust, amely eldönti, van-e két olyan sor, amelyeknek az első oszlopbeli elemei különböznek, viszont az összes többi oszlopban megegyeznek.