

Keresés, rendezés

Algoritmuselmélet

2019. tavasz

12. gyakorlat

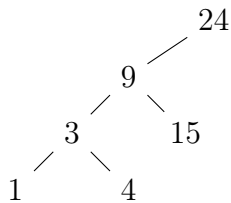
Piros-fekete fa

A piros-fekete fa egy olyan bináris fa, melyre teljesülnek a következők.

1. Minden nemlevél (belső) csúcsnak 2 fia van.
2. Az elemeket a belső csúcsokban tároljuk.
3. Teljesül a keresőfa tulajdonság.
4. A fa minden csúcsa piros vagy fekete.
5. A gyökér fekete.
6. A levelek feketék.
7. Minden piros csúcs mindkét gyereke fekete.
8. Minden v csúcsra igaz, hogy az összes v -ből levélbe vezető úton ugyanannyi fekete csúcs van.

1. Rendezzük az 5, 2, 10, 7, 1, 8 tömböt (a) buborék-, (b) beszúrásos, (c) összefésüléses, (d) gyorsrendezéssel! Hány összehasonlítást végeztünk?
2. Rendezzük a $cab, abb, cca, bcb, cbc, aac, bac$ elemeket radix rendezéssel.
3. Adott az $A[1..n]$ csupa különböző egész számot növekvő sorrendben tartalmazó tömb. Adjunk hatékony algoritmust egy olyan i index meghatározására, melyre $A[i] = i$, feltéve, hogy van ilyen i .
4. Az $A[1..n]$ tömbben levő elemekről tudjuk, hogy $A[1] \neq A[n]$. Adjunk $O(\log n)$ összehasonlítást használó algoritmust, amely talál egy olyan i indexet, hogy $A[i] \neq A[i + 1]$.
5. Egy tömbben n elemet tárolunk. 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.
6. Adottak az $a_1, a_2, \dots, a_n$ (nem feltétlenül csak pozitív) 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. Adjon erre a feladatra egy $O(n)$ összehasonlítást használó algoritmust!
7. Adott az n elemet tároló A tömb. Hogyan lehet $O(n \log n)$ összehasonlítással találni egy olyan $i \neq j$ indexpárt, amire $|A[i] - A[j]| < 100$ teljesül?
8. Az $A[1..n]$ egy természetes számokból álló tömb. Szeretnénk ellenőrizni, hogy az elemek rendezetten vannak-e tömbben. Olyan algoritmust adjunk, ami minimális számú összehasonlítást használ. Hány összehasonlítást használ egy ilyen algoritmus? (Azt is be kell látni, hogy nincs ennél kevesebb összehasonlítást használó algoritmus. Azt nem kell eldöntenie az algoritmusnak, hogy növekvő, vagy csökkenő sorrendben vannak-e rendezve.)
9. Egy tányérra n különböző átmérőjű, kör alakú palacsinta van felhalmozva véletlenszerű sorrendben. Szeretnénk a palacsintákat alulról felfelé növekvő sorrendbe rendezni. Ehhez egyféle műveletet használhatunk: egy lapáttal benyúlunk bármely palacsinta alá és a felette levő kupacot fejjel lefelé fordítva visszatesszük a kupacra. Adjunk olyan algoritmust, ami minden esetben $O(n)$ művelettel elvégzi a feladatot!
10. Az $A[0..n]$ tömbre teljesül, hogy minden $2 \leq i \leq n - 2$ esetén $A[i - 2] < A[i] < A[i + 2]$. Adjon a tömb rendezésére egy $O(n)$ összehasonlítást használó algoritmust!

11. Tudjuk, hogy az $a_1, \dots, a_n$ sorozat olyan, hogy egy darabig növekszik, utána csökken. Adjunk $O(n)$ összehasonlítást használó algoritmust, ami növekvő sorrendbe rendezi az elemeit!
12. Az $a_1, \dots, a_n$ természetes számokról tudjuk, hogy 10 darab kivétellel teljesül, hogy $10 \leq a_i \leq 10n$. A kivételes 10 természetes szám bármekkora lehet. Adjunk $O(n)$ futásidejű algoritmust, ami növekvően rendezi az összes számot!
13. Vázzunk egy $O(n)$ időigényű algoritmust (az időkorlát bizonyításával együtt) n olyan egész számból álló sorozat rendezésére, melynek elemei az
 - (a) $\{1, \dots, 3n\}$ tartományba esnek!
 - (b) $\{1, \dots, n^7 - 1\}$ tartományba esnek!
14. Igazoljuk, hogy nincs olyan összehasonlításokat használó rendezőalgoritmus, amely egy tetszőleges $A[1..n]$ tömb rendezésekor az $A[1]$ elemet minden másikkal összehasonlítja, az $A[2]$ elemet legfeljebb $\lceil n/2 \rceil$ elemmel, és általában az $A[i]$ elemet legfeljebb $\lceil n/2^{i-1} \rceil$ elemmel hasonlítja össze!
15. Építsünk beszúrásokkal bináris keresőfát az alábbi sorrendben érkező számokból: 10, 3, 8, 12, 1, 5, 15, 4, 6, 13. Hajtsuk végre a TÖRÖL(12) és TÖRÖL(10) műveleteket.
16. Az ábrán egy bináris keresőfa látható, melyet naiv beszúrásokkal építettünk. Milyen sorrendben végezhetjük a beszúrásokat? Hányféle lehetséges sorrend van?



17. Egy 7 csúcsú bináris fa preorder bejárásakor a csúcsok sorrendje: A, B, C, D, E, F, G , inorder bejárásakor pedig: B, E, D, F, C, G, A . Rajzoljuk fel, hogyan nézhet ki a fa!
18. Az A bináris keresőfában n egész számot, a B -ben pedig m -et tárolunk. Rendezzük az $n + m$ elemet $O(n + m)$ lépésben!
19. Egy piros-fekete fában az x csúcs gyermekei y_1 és y_2 , az y_2 gyermekei z_1, z_2 . Tudjuk még, hogy y_1 levél. Mi mondható x, y_1, y_2, z_1, z_2 színéről?
20. Mennyi a tárolható elemek számának minimuma, illetve maximuma egy olyan piros-fekete fában, aminek a fekete magassága 3?
21. Egy piros-fekete fa fekete magassága 5, és tudjuk, hogy egyetlen piros csúcsa van. Legalább, illetve legfeljebb hány elemet tárolhat a fa?
22. Egy piros-fekete fában jelölje x és y a gyökér két fiát. Tudjuk, hogy $\text{fm}(x) = \text{fm}(y)$, de az x csúcs két gyerekének különbözik a fekete magassága. Milyen színű az y csúcs?
23. Egy piros-fekete fában a gyökérnek és még egy csúcsnak 100 a fekete magassága, az összes többi csúcs fekete magassága ennél kisebb. Határozza meg a fában tárolható elemek minimális számát!