Kupacok, bináris keresőfák

Algoritmuselmélet

7. gyakorlat

2024.

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. Adjunk hatékony algoritmust egy kupac tizedik legkisebb elemének megtalálására.
4. Igazoljuk, hogy egy n elemből álló kupac felépítése $\Omega(n)$ összehasonlítást igényel.
5. Bizonyítsuk be, hogy egy n elemből álló kupac legnagyobb elemének megkereséséhez $\Omega(n)$ összehasonlítás szükséges.
6. Adott egy n elemet tartalmazó kupac és egy K kulcs. Keressük meg a kupac K-nál kisebb kulcsú elemeit; ha m ilyen elem van, akkor az algoritmus $O(m)$ elemi lépést használhat.
7. Egy n-elemű kupac egyik elemét megváltoztattuk. Feltéve, hogy a változtatás helye ismert, adjunk $O(\log n)$ lépésszámú algoritmust a kupac-tulajdonság helyreállítására.

   (a) Építsünk beszúrásokkal bináris keresőfát az alábbi sorrendben érkező számokból: 10, 3, 8, 12, 1, 5, 15, 4, 6, 13.
   (b) Hajtsuk végre rendre a TÖRÖL(4), TÖRÖL(12) és TÖRÖL(10) műveleteket.
   (c) Milyen sorrendben írja ki a preorder, inorder és posztorder bejárás a fában tárolt értékeket?
8. Egy bináris keresőfában csupa különböző egész számot tárolunk. Lehetséges-e, hogy egy KERES(x) hívás során a keresési út mentén a 20, 18, 3, 15, 5, 8, 9 kulcsokat látjuk ebben a sorrendben?
9. Egy bináris fa inorder bejárása $j, b, k, g, i, a, c, d, f, e, h$, preorder bejárása pedig $a, b, j, g, k, i, d, c, e, f, h$. Rekonstruáljuk a fát.
10. Egy bináris keresőfában n egész számot, egy másikban pedig m egész számot tárolunk. Rendezzük az $n + m$ elemet $O(n + m)$ lépésben.
11. Egy bináris keresőfában n különböző egész számot tárolunk. Adjunk algoritmust, ami $O(n)$ lépésben eldönti, hogy van-e a tárolt számok között két olyan, melyek különbsége 2013.
12. Adott két bináris keresőfa, mindegyikben n különböző elemet tárolunk. Adjunk $O(n)$ lépésszámú eljárást, ami eldönti, hogy igaz-e, hogy a két fában ugyanazok a számok szerepelnek.
13. Adott két teljes bináris keresőfa, mindegyikben n elemet tárolunk. Adjunk $O(\log n)$ lépésszámú eljárást, ami eldönti, hogy az első fa összes eleme nagyobb-e a második fa összes eleménél.
14. Adott egy n-csúcsú bináris keresőfa, ami csupa különböző elemet tárol. Ennek minden v csúcsára meg akarjuk határozni, hogy a v gyökerű részfában hány darab v-nél kisebb elem van tárolva. Adjunk algoritmust, ami ezt a feladatot $O(n)$ lépésben megoldja.

15. Egy orvosi rendelőben a regisztrációnál kell bejelentkezni, ahol az ott dolgozók eldöntik, hogy a beteg az épp rendelő két orvos közül A-hoz vagy B-hez kell kerüljön, vagy bármelyikükhöz kerülhet. Ezen kívül, a beutaló ismeretében, a beteghez egy, a sürgősséget kifejező, számot is rendelnek. Amikor valamelyik orvos végzett egy beteggel, akkor azon betegek közül, akiket nem csak a másik orvos láthat el, behívja a legkisebb sürgősségi számút. Tegyük fel, hogy a kiosztott sürgősségi számok egymástól különbözőek. Írjunk le egy olyan adatszerkezetet, ami abban az esetben, ha n beteg várakozik, akkor a regisztráción az új beteg beillesztését, illetve az orvosoknak a következő beteg kiválasztását $O(\log n)$ lépésben lehetővé teszi.

16. Adott n pont a síkon, melyek páronként mindkét koordinátájukban különböznek. Bizonyítsuk be, hogy pontosan egy olyan bináris fa létezik az n pont tárolására, mely a pontok első koordinátája szerint keresőfa tulajdonsággal, a második szerint pedig kupac tulajdonsággal rendelkezik. (Vigyázat: a kupac tulajdonságba nem értendő bele, hogy a fa teljes bináris fa legyen.)