

1. (a) Építsen beszúrásokkal bináris keresőfát az alábbi sorrendben érkező számokból:  
7, 3, 2, 9, 8, 12, 6, 4.  
(b) Járja be pre-, in-, és posztorder bejárással a kapott fát!  
(c) Az (a) rész keresőfáján hajtsa végre az alábbi műveletsort: BESZÚR(5), TÖRÖL(2), TÖRÖL(7), TÖRÖL(6).
  2. (a) Építsen kupacot az órán tanult lineáris idejű módszerrel az alábbi tömbből: 31, 6, 50, 7, 2, 51.  
(b) Szúrja be az így kapott tömbbe az 1, majd ezután az 5 számot.  
(c) Hajtson végre két egymást követő MINTÖR-t az így kapott kupacon.
- 
3. Bizonyítsuk be, hogy ha az inorder bejárást egy bináris keresőfára futtatjuk, akkor az elemeket rendezett sorrendben kapjuk meg.
  4. (a) Építsen kupacot az órán tanult lineáris idejű módszerrel az alábbi tömbből: 4, 3, 5, 21, 2, 7, 12, 6.  
(b) Szúrja be az így kapott tömbbe az 1 számot.  
(c) Hajtson végre két egymást követő MINTÖR-t az így kapott kupacon.
  5. Igazoljuk, hogy egy  $n$  elemből álló bináris kupac felépítése bármilyen a tanult kupacépítés módszerétől eltérő módszerrel is  $\Omega(n)$  összehasonlítást igényel!
  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, az algoritmus  $O(m)$  elemi lépést használhat.
  7. Igazolja, hogy minden olyan algoritmus, ami csak összehasonlításokkal fel tud építeni egy bináris keresőfát  $n$  elem esetén  $\Omega(n \log n)$  összehasonlítást használ!
- 
8. A kezdetben üres kupacba egyenként szúrunk be  $n$  elemet. Igazolja, hogy előfordulhat, hogy a beszúrások során végzett összehasonlítások száma  $\Omega(n \log n)$ .
  9. Egy halmazból  $n$  elem kupacban van elhelyezve, bármely kettő összehasonlítható. Bizonyítsuk be, hogy a legnagyobb elem megkereséséhez  $\Omega(n)$  összehasonlítás szükséges!
  10. Adjunk hatékony algoritmust egy kupac tizedik legkisebb elemének a megtalálására. Elemezzük a módszer költségét.
  11. 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 sürgősséget kifejező számot is rendelnek. Amikor valamelyik orvos végzett egy beteggel, akkor a betegek közül, akiket nem csak a másik orvos láthat el behívja a legnagyobb sürgősségi számút. Tegyük fel, hogy a kiosztott sürgősségi számok egymástól különbözőek. Írjon 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 és az adatstruktúra frissítését  $O(\log n)$  lépésben lehetővé teszi.
  12. Egy halmaz adott  $n$  eleméből (bármely két elem összehasonlítható) kupacot szeretnénk építeni, melyre (a kupac tulajdonság mellett) teljesül, hogy minden  $x$  csúcs bal részfájának az elemei kisebbek mint az  $x$  jobb részfájának az elemei. Igazoljuk, hogy ehhez legalább  $\Omega(n \log n)$  lépés szükséges!
  13. Adott  $n$  pont a síkon, melyek páronként mindkét koordinátájukban különböznek. Bizonyítsuk be, hogy egy és csak egy bináris fa létezik, melynek csúcsaiban az adott  $n$  pontot tároljuk, és az első koordináta szerint a keresőfa tulajdonsággal, a második szerint pedig a kupac tulajdonsággal rendelkezik. (*Vigyázat:* a kupac tulajdonságba nem értendő bele, hogy a fa teljes bináris fa legyen, mint amilyet a tanult „kupacépítő” algoritmus létrehoz.)
  14. 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? Ha nem lehetséges, indokolja meg miért nem, ha pedig lehetséges, határozza meg az összes olyan  $x$  egész számot, amire ez megtörténhet.