

1. Építsünk kupacot a következő tömbből: [4, 11, 9, 10, 5, 6]. Ezután szűrjük be a következő számokat: 8, 1, 2, 16. Levezetésként csináljunk 3 MINTÖR-t!
 2. **[ZH: 2008. március 28.]** 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 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 $O(\log n)$ lépésben lehetővé teszi.
-
3. **[PZH: 2008. május 9.]** A 10 elemű A tömb első 8 elemére legyen $A[i] = 2i (1 \leq i \leq 8)$, és tekintsük ezt, mint egy 8 elemű kupacot. Rajzolja le az ehhez tartozó fát! Hajtsa végre rajta a BESZŰR(3), BESZŰR(1), MINTÖR műveletsort, rajzolja le az egyes műveletek után a kupacot (és jelezze a közben szükséges részlépéseket is)!
 4. **[ZH: 2010. április 19.]** Igaz-e, hogy az $A[1] = 3, A[2] = 15, A[3] = 10, A[4] = 25, A[5] = 29, A[6] = 17, A[8] = 28, A[9] = 30$ tömb egy kupacot tartalmaz? Ha igen, rajzolja le a kupacot és a rajzon hajtsa végre a BESZŰR(11) műveletet!
 5. Csináljuk meg az 1. feladatot 4-kupaccal!
 6. **[Vizsga: 2008. május 27.]** Egy kupacba beraktunk egy új x elemet, majd végrehajtottunk egy MINTÖR műveletet. Mikor fordul elő, hogy végül az eredeti kupacot kapjuk vissza?
 7. Adjunk hatékony (*hehe* :) algoritmust egy kupac tizedik legkisebb elemének meghatározására!
 8. Adott egy n elemet tartalmazó kupac és egy k kulcs. Keressük meg a kupac k -nál kisebb elemeit! Ha m ilyen elem van, akkor az algoritmus $O(m)$ elemi lépést használhat.
 9. Adjunk konstans szorzó erejéig optimális költségű algoritmust az alábbi problémára!
INPUT: Egy $A[1 : n]$ tömb, amely eredetileg az $1, \dots, n$ számokat tartalmazta kupacba rendezve, de öt elem megsérült, és a helyére $*$ került.
FELADAT: Találjuk meg a tömb összes olyan kitöltését, ami lehetett az eredeti!
 10. Tervezzünk olyan adatszerkezetet, ami egy rendezett halmaz elemeinek tárolására szolgál. A megvalósítandó műveletek: *Felépít*(n): n elemből felépíti a struktúrát; *Mintör*, *Maxtör*: a min. illetve max. elem törlése; *Beszúr*(x): az x elemet a struktúrába illeszti. Az egyes műveletek költsége ne legyen több, mint *Felépít*: $O(n)$; *Mintör*, *Maxtör*, *Beszúr*: $O(\log n)$, ahol n a tárolt elemek száma.