

4. gyakorlat

Kupac adatszerkezet - Keresések

0. Mi korlátozza az adjacenciamátrixával adott G gráfban a legrövidebb utak számítási költségét a Bellman-Ford, a Floyd és a Dijkstra algoritmus esetén? Azaz miért $O(n^3)$ ill. $O(n^2)$? Honnan jönnek az n -ek és elképzelhető-e speciális eset, amikor a kitevő biztosan kisebb?
1. Adott az $A[1 : n]$ csupa különböző egész számot növekvő sorrendbentartalmazó tömb. (A tömbben negatív számok is lehetnek!) Adjunk hatékony algoritmust egy olyan i index meghatározására, melyre $A[i] = i$ (feltéve, hogy van ilyen i): igyekezzünk minél kevesebb elem megvizsgálásával megoldani a feladatot!
2. Adott az $A[n]$ csupa különböző számot tartalmazó tömb. Tudjuk, hogy $A[1] < A[2]$ és $A[n-1] > A[n]$. Adjunk $O(\log n)$ költségű algoritmust, amely megtalál egy lokális maximumot az A tömbben, ha van ilyen.
3. Építsünk kupacot a 10, 8, 6, 1, 3, 2, 5, 7, 9, 4 sorozatból!
4. (a) Építsünk kupacot az órán tanult lineáris idejű módszerrel az alábbi tömbből: 31; 6; 50; 7; 2; 51.
(b) Szűrjük be az így kapott tömbbe az 1, majd ezután az 5 számot.
(c) Hajtsunk végre két egymást követő MINTÖR-t az így kapott kupacon.
5. Adott egy n hosszú, csupa különböző elemből álló lista. Hogyan választhatjuk ki hatékonyan az öt legkisebbet? És az ötödik legnagyobb? Mennyire hatékonyan? Ha a listából épített kupac adott, akkor milyen költséggel tudjuk ugyanezeket megtenni?
6. Egy rendezett halmazból n elem kupacban van elhelyezve. Bizonyítsuk be, hogy a legnagyobb elem megkereséséhez $\Omega(n)$ összehasonlítás szükséges!
7. A kezdetben üres kupacba egyenként szúrunk be n elemet. Igazoljuk, hogy előfordulhat, hogy a beszúrások során végzett összehasonlítások száma nem $O(n)$.
8. 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. 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.
9. Ha adott n szám, akkor hívjuk közülük középső elemnek a rendezés szerinti $\lceil n/2 \rceil$ -ediket. Kezdetben adottak az $a_1, a_2, \dots, a_n$ egész számok, amikről tudjuk, hogy az a_1 a középső elem, egyébként a számok rendezetlenek. Ezekből építsünk fel egy adatszerkezetet, amiben két művelet van:
BESZÚR: egy új elemet illeszt az adatszerkezetbe,
KÖZÉPTÖR: az aktuális középső elemet törli.
Mindkét művelet megvalósítása $O(\log k)$ összehasonlítást használjon, amikor k tárolt elem van, az adatszerkezet kezdeti felépítése legyen $O(n)$ összehasonlítás.
10. Adjunk hatékonyan használható adatstruktúrát arra, ha egy számlista harmadik legkisebb és harmadik legnagyobb elemét felhasználva akarjuk eldönteni, hogy töröljük-e a legkisebb elemet. A tárolt számlistára legyen igaz az, hogy a harmadik legkisebb és a harmadik legnagyobb relatív prímek. Az eredeti listából a megfelelő listát úgy kapjuk, hogy a feltételnek való ellentmondás esetén töröljük listából a legkisebb elemet, a harmadik legkisebb elem ezáltal megváltozik. A struktúra felépítési költsége $O(n)$ legyen. Mennyi a törlések maximális száma és egy nem megfelelőség mit von maga után?