

6. gyakorlat

Keresések és rendezések

- Vidéken autózunk, ahol benzinkút csak bizonyos falvakban van. Az A falubeli benzinkúttól indulunk és a B faluba akarunk elérni (ahol szintén van benzinkút). A falvak közötti utakat egy n csúcsú e élű, összefüggő, irányítatlan gráf írja le, melynek csúcsai a falvak, az élek pedig a falvak közötti utakat jelentik, egy él súlya a két falut összekötő útszakasz hossza. A gráf az éllistájával adott, és ezen kívül adott még az a k falu, amelyben van benzinkút. Adjunk $O(k \cdot e \cdot \log n)$ lépésszámú algoritmust, amely meghatározza az A -ból B -be vivő legrövidebb olyan útvonalat, melynek során soha nem kell 600 kilométernél többet autóznunk két benzinkút között.
- 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.
- 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.
- Hogyan működik a beszúrásos, az összefésüléses, a buborék- és a gyorsrendezés eljárás? Milyen lépésszámmal végezhető el egy n elemű lista rendezése ezekkel az algoritmussal?
- Rendezzük a következő listát összefésüléses rendezés segítségével: 4, 11, 9, 10, 5, 6, 8, 1, 2, 16.
- Egy csupa különböző egészekből álló sorozat *bitonikus*, ha először nő, utána pedig fogy, vagy fordítva: először fogy, utána nő. Például az (1, 3, 7, 21, 12, 9, 5), (9, 7, 5, 4, 6, 8) és (1, 2, 3, 4, 5) sorozatok bitonikusak. Adjunk $O(n)$ összehasonlítást használó rendező algoritmust n elemű bitonikus sorozatok rendezésére!
- Az $A[1 : n]$ tömbben egy rendezett univerzum n különböző eleme volt, nagyság szerint növekvő sorrendben. Valaki időközben megkeverte a tömb elemeit, de csak annyira, hogy minden egyes elem új helye az eredetitől legfeljebb 5 távolságra esik. Adjunk $O(n)$ idejű algoritmust az eredeti állapot helyreállítására!
- Adott az $A[1 : n]$ csupa különböző egész számot növekvő sorrendben tartalmazó 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!
- Adott egy dobozban n különböző méretű anyacsavar, valamint egy másik dobozban a hozzájuk illő apacsavarok. Kizárólag a következő összehasonlítási lehetőségünk van: Egy apacsavarhoz hozzápróbálunk egy anyacsavart. A próbának háromféle kimenete lehet: $\text{apa} < \text{anya}$, $\text{apa} = \text{anya}$, vagy $\text{apa} > \text{anya}$; annak megfelelően, hogy az apacsavar külső átmérője hogyan viszonyul az anyacsavar belső átmérőjéhez. Szeretnénk az anyacsavarokhoz megtalálni a megfelelő apacsavarokat. Adjunk erre a feladatra **átlagosan** $O(n \log n)$ összehasonlítást felhasználó módszert!
- Hogyan működik a láda- és a (lexikografikus) radix-rendezés?
- Rendezzük radix rendezéssel az acedf, adfce, ecdafe, acfde, edcaf, edcfa, caedf elemeket.
- Egy azonos hosszú abc-szerint rendezett (ismétlődő elem nélküli) karaktersorozatból álló lista kolerikografikusan rendezett, ha bármely két i és j elem esetén i pontosan akkor előzi meg j -t, ha az i lexikografikus ("normál") rendezés szerinti utolsó karaktere megelőzi j lexikografikusan utolsó elemét. (Pl.: *abc, abd, acd, bcd, abe, ace, ade, bce, ...*) Hogyan rendezhetnénk sorba a feltételnek megfelelően azonos hosszú, abc-szerint rendezett karaktersorozatokat. Mennyi ennek a költsége? Rendezzük eszerint ezeket: *cdf, bcf, ace, cef, cde, bcd*.
- Vázoljunk egy $O(n)$ időigényű algoritmust (az időkorlát bizonyításával együtt) n olyan egész számból álló sorozat rendezésére, melynek elemei az
 - $\{1, \dots, 3n\}$ tartományba esnek!
 - $\{1, \dots, n^7 - 1\}$ tartományba esnek!