Algoritmuselmélet
2011. április 6.

Schlotter Ildi
ildi@cs.bme.hu

Gyakorló feladatok ZH-ra

Nagyságrendek

1. Egy algoritmusról tudjuk, hogy a lépésszáma $O(n^2)$. Lehetséges-e, hogy
(a) minden páros n-re az n hosszú bemeneteken a lépésszám legalább $2008n \log^3 n$;
(b) minden n hosszú bemeneten a lépésszám legfeljebb $3^{\log n}$?

2. Tegyük fel, hogy $f(n) = O(g(n))$. Következik-e ebből, hogy $4^{f(n)} = O(2^{g(n)})$?

3. Az alábbi függvényeket rendezze olyan sorozatba, hogy ha f_i után közvetlenül f_j következik, akkor $f_i = O(f_j)$ teljesüljön!

$$f_1 = \log(100n)^2 \quad f_2 = 113n + 2(\log n)^{\log n} \quad f_3 = 52(\log(64n))^2$$

4. Az $\mathcal{A}$ algoritmus lépésszáma az n hosszú bemeneteken legyen legfeljebb $L(n)$. Tudjuk, hogy $L(n) \leq L(n-1) + n$ teljesül, ha $n > 1$ és $L(1) = 1$. Igazolja, hogy $L(n) = O(n^2)$.

5. Van egy számítógépes programunk, ami egy k méretű feladaton a jelenlegi gépünkön 1 nap alatt fut le. Beszereztünk egy százszor gyorsabb gépet. Ugyanazon programmal mekkora feladatot lehet az új gépen 1 nap alatt megoldani, ha a program lépésszáma n méretű feladat esetén
(a) n-nel,
(b) n^3-bel,
(c) 2^n-nel arányos?

6. Egy $\mathcal{A}$ algoritmusról azt tudjuk, hogy az n hosszú bemeneteken a lépésszáma $O(n \log n)$. Lehetséges-e, hogy
(a) van olyan x bemenet, amin a lépésszáma $|x|^3$?
(b) minden x bemeneten legfeljebb $2007|x|$ lépést használ?
(Szokás szerint $|x|$ az x szó hosszát jelöli.)

7. Igaz-e, hogy
(a) ha $f = O(g)$ és $g = O(h)$, akkor $f = O(h)$?
(b) ha $f = \Omega(g)$ és $g = \Omega(h)$, akkor $f = \Omega(h)$?

8. Az $\mathcal{A}$ algoritmusról azt tudjuk, hogy összefüggő gráfokon $O(n+e)$ lépést tesz. Mutassa meg, hogy az is igaz, hogy összefüggő gráfokon az algoritmus lépésszáma $O(e)$.

9. Jelölje egy algoritmus maximális lépésszámát az n hosszú bemeneteken $L(n)$. Azt tudjuk, hogy minden $n = 2k > 4$ páros számra $L(2k) \leq L(2k-2) + 1$ teljesül, és hogy $L(4) = 10$. Következik-e ebből, hogy az algoritmus lépésszáma $O(n)$?

Keresés, rendezés

1. Hány összehasonlítással lehet egy n elemű tömbben a legkisebb elemet megtalálni?

2. Adjunk olyan algoritmust, amely $\lceil 1.5n \rceil - 2$ összehasonlítással megtalálja egy n elemű tömb legnagyobb és legkisebb elemét!

3. 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!

4. Az egész elemeket tartalmazó $A[1:n]$ tömböt konvexnek nevezzük, ha minden i-re $(1 < i < n)$ teljesül, hogy $A[i] \leq 1/2(A[i-1] + A[i+1])$. Javasoljunk olyan algoritmust, mely minél kevesebb összehasonlítással megtalálja egy konvex tömb minimális elemét!

5. 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!

6. Rendezze az $7, 3, 12, 1, 5, 4$ tömböt
   a) buborékrendezéssel,
   b) beszúrásos rendezéssel,
   c) összefésüléses rendezéssel,
   d) kupacos rendezéssel!

7. Adott egy egész számokat tartalmazó $A[1..n]$ tömb, amelyben legfeljebb n elempár áll inverzióban egymással (két elem akkor áll inverzióban, ha a nagyobb megelőzi a kisebbet). Igaz-e, hogy a buborék-rendezés rendezi az A tömböt
   a) legfeljebb n összehasonlítással?
   b) legfeljebb n cserével?

8. Az n méretű (nem feltétlenül rendezett) A tömb elemei különböző pozitív egész számok. Adjon algoritmust, amely meghatároz egy $1 \leq k \leq n$ számot és kiválaszt k különböző elemet az A tömbből úgy, hogy a kiválasztott elemek összege nem több mint k^3. Ha nincs ilyen k, akkor az algoritmus jelezze ezt a tényt. Az algoritmus lépésszáma legyen $O(n \log n)$. (Két szám összehasonlítása, összeadása vagy szorzása egy lépésnek számít.)

9. Rendezzük a következő láncokat a radix rendezés segítségével: $abc,\ \ acb,\ \ bca,\ \ bbc,\ \ acc,\ \ bac,\ \ baa$.

10. Legyen adott egy egészekből álló $A[1:n]$ tömb és egy b egész szám. Szeretnénk eldönteni, hogy van-e két olyan $i, j \in \{1, \ldots, n\}$ index, melyekre $A[i] + A[j] = b$. Oldjuk meg ezt a feladatot $O(n \log n)$ időben!

11. Az $A[1:n]$ tömbben egész számokat tárolunk, ugyanaz a szám többször is szerepelhet. Határozzuk meg $O(n \log n)$ lépésben a leggyakoribb számokat, vagyis azokat, amelyeknél többször semelyik másik szám sem fordul elő a tömbben.

12. Vázoljunk egy $O(n)$ időigényű algoritmust (az időkorlát bizonyításával együtt) n olyan egész számból álló sorozat rendezésére, melynek elemei az $\{1, \ldots, 3n\}$ tartományba esnek!

13. A 4 elemű I abc felett adott két szó: $x = x_1 x_2 \cdots x_n$ és $y = y_1 y_2 \cdots y_k$, ahol $1 \leq k \leq n$ és $x_i, y_j \in I$. Keressük az x szóban az olyan részszavakat, amelyek anagrammái y-nak, azaz az olyan i indexeket, hogy az $x_i, x_{i+1}, \ldots, x_{i+k-1}$ betűk megfelelő sorrendbe rakva az y szót adják. Adjon algoritmust, ami x-ben az összes ilyen i helyet $O(n)$ lépésben meghatározza.

Bináris keresőfák, 2-3 fák, B-fák

1. Építsen beszúrásokkal bináris keresőfát az alábbi sorrendben érkező számokból:
   $8, 1, 3, 10, 2, 15, 9, 6, 5, 12, 2, 13$.
   a) Törölje ki az $1, 9$ és 8 elemeket!
   b) Milyen sorrendben írja ki a preorder, inorder és posztorder bejárás a csúcsokat?

2. Adott egy n csúcsú és egy k csúcsú bináris keresőfa. A két fában tárolt összes elemből $O(n + k)$ lépésben készítsen egy rendezett tömböt!

3. Határozza meg azokat a bináris fákat, amikben a preorder bejárás szerinti sorrend éppen a postorder bejárás által adott sorrend fordítottja!

4. Egy bináris keresőfában $1, 2, \ldots, 100$ számokat tároljuk. A baloldali részfa 16 elemet tárol. Mi lehet a gyökérben lévő elem? Minimum és maximum mekkora lehet a bal- illetve a jobboldali részfák magassága?

5. Egy bináris keresőfa "valamely bejárásán" mindig a $\{pre,\ in,\ post\}$-order valamelyikét értjük.
   a) Mely bejárásoknál lehetséges az, hogy a tárolt elemek legnagyobbika megelőzi a legkisebbet?
   b) Tegyük fel, hogy egy bináris keresőfában az $1, 2, \ldots, n$ számok vannak tárolva, továbbá hogy a fa valamely bejárásánal a számok az $n, n-1, \ldots, 1$ sorrendben következnek. Határozzuk meg, melyik lehetett ez a bejárás és milyen lehetett ez a bináris keresőfa!

6. 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.

7. Adjuk meg azt a bináris fát, melynek inorder és postorder bejárásai a következőt adják:
   Postorder: D, A, H, E, F, G, B, C.
   Inorder: A, D, C, H, F, E, B, G.

8. Adott n pont a síkon, melyek páronként mindkét koordinátájukban különböznek. Bizonyítsuk be, hogy egy és csak egy bináris fa létezik, melynek pontjai az adott n pont, és az első koordináta szerint a keresőfa tulajdonsággal, a második szerint pedig a kupac tulajdonsággal rendelkezik.

9. Vázolja egy ismert adatszerkezet olyan módosítását, ami lehetővé teszi, hogy amikor n elemet tárolunk benne, akkor a KERES, BESZÚR, TÖRÖL műveletek lépésszáma $O(\log n)$, a MIN, MAX lépésszáma pedig $O(1)$.

10. Írjon le egy olyan adatszerkezetet, amivel egész számok véges részhalmazait tárolhatjuk. Jelölje T_i ($i = 1, \ldots, n$) a tárolandó halmazokat. Három műveletet definiálunk,
    BESZÚR(i,x): a T_i halmazhoz hozzáveszi az $x \in A$ elemet
    METSZETMÉRET(i,j): megadja a két halmaz metszetének $|T_i \cap T_j|$ elemszámát
    UNIÓMÉRET(i,j): megadja a két halmaz uniójának $|T_i \cup T_j|$ elemszámát.

    A BESZÚR lépésszáma legyen $O(|T_i|)$, a másik két műveleté pedig $O(|T_i| + |T_j|)$.

11. Építsen 2-3 fát az alábbi sorrendben érkező számokból: $1, 2, 3, 4, 5, 6, 7$. A felépített fából törölje az $1, 5, 6$ számokat.

12. Illesszük be az alábbi 6 kulcsot egy kezdetben üres 2-3 fába a megadott sorrendben: D, B, E, A, C, F. Rajzoljuk le az eredményül kapott fát!

13. Egy 2-3 fába egymás után 1000 új elemet illesztettünk be. Mutassa meg, hogy ha ennek során egyszer sem kellett csúcsot szétvágni, akkor a beillesztések sorozata előtt már legalább 2000 elemet tároltunk a fában.

14. Egy kezdetben üres 2-3-fába az $1, 2, \ldots, n$ számokat szúrtuk be ebben a sorrendben. Bizonyítsa be, hogy a keletkezett fában a harmadfokú csúcsok száma $O(\log n)$.

15. Egy 2-3 fában egy rendezett halmaz 10 000 elemét tároljuk. Milyen korlátok közé esik a fa magassága?

16. Egy 2-3 fa magassága 8. Mennyi az fában tárolt elemek minimális illetve maximális száma?

17. F egy B_{12}-fa, melynek a magassága 8. Mennyi az F-ben tárolt elemek minimális illetve maximális száma?

18. Egy B_m-fában rekordokat szeretnénk tárolni. Egy rekord hossza 100 byte, egy kulcs hossza 20 byte, egy mutató hossza pedig 4 byte. A lapméret 1000 byte. Mekkorára válasszuk m-et, hogy bármelyik belső csúcs ráférjen egy lapra? Legalább hány szintje lesz az így kapott fának, ha 20 millió rekordot szeretnénk tárolni? (A tárolás során egy levélben több rekordot is tárolhatunk, ha azok ráférnek egy lapra.)

19. Adott egy $n = 2^k - 1$ pontú teljes bináris keresőfa. A fában tárolt elemek egészek az $I = [1, 2^k]$ intervallumból és egy szám legfeljebb egyszer fordul elő a fában. Utóbbi feltétel szerint pontosan egy olyan $i \in I$ egész van, amely nincs a fában. Adjunk egy hatékony módszert i meghatározására.

20. Tervezzen adatstruktúrát a következő feltételekkel. Természetes számokat kell tárolni, egy szám többször is szerepelhet. A szükséges műveletek:
    BESZÚR(i): i egy újabb példányát tároljuk
    TÖRÖL(i): i egy példányát töröljük
    MINDTÖRÖL(i): i összes példányát töröljük
    DARAB(i): visszaadja, hogy hány példány van i-ből
    ELEM(K): megmondja, a nagyság szerinti rendezésben a K-adik elem értékét.
    Az adatstruktúra legyen olyan, hogy ha m-*féle* elemet tárolunk, akkor mindegyik művelet lépésigénye $O(\log m)$. (Például ha a tárolt elemek 1,1,3,3,3,8, akkor DARAB(1)=2, ELEM(4)=3 és m =3.)

Hashelés

1. A hash-függvény legyen $f(K) = K \pmod M$, a táblaméret $M = 7$, és $1 \le K \le 20$. Helyezzük el a táblában a 3, 4, 7, 11, 14, 17, 20 kulcsokat ebben a sorrendben, majd töröljük a 11, 17 elemeket, és végül szúrjuk be a 10 kulcsot. Összesen hány ütközés történt, és mi a tábla végső állapota, ha
   a) lineáris
   b) kvadratikus maradék próbálást használtunk az ütközések feloldására?

2. Nyitott címzéssel hashelünk egy kezdetben üres $M = 11$ méretű táblába, $h(K) = K \pmod M$ hash-függvénnyel. Az ütközések feloldására kettős hashelést használunk, ahol a második hash-függvény $h'(K) = (7K \pmod{(M-1)})$. Mi lesz a tábla állapota, ha a 4, 5, 15, 7, 16, 21, 26 kulcsokat szúrjuk be ebben a sorrendben?

3. A $T[0:M]$ táblában $2n$ elemet helyeztünk el az első $3n$ helyen ($3n < M$) egy ismeretlen hash-függvény segítségével. A táblában minden $3i$ indexű hely üresen maradt ($0 \leq i < n$). Legfeljebb hány ütközés lehetett, ha az ütközések feloldására
a) lineáris próbálást
b) kvadratikus maradék próbálást használtunk?

4. Előfordulhat-e nyitott címzéses hash-elés esetén, hogy az $n > 3$ méretű táblában csak 3 elem van, de a keresés lépésszáma n?

5. Egy m méretű hash-táblában már van néhány elem. Adjon $O(m)$ lépésszámú algoritmust, amely meghatározza, hogy egy újabb elem lineáris próbával történő beszúrásakor maximum hány ütközés történhet.

6. A $T[0:M-1]$ táblában rekordokat tárolunk nyitott címzésű hashelt szervezéssel. Az ütközések feloldására lineáris próbálást alkalmazunk. Tegyük fel, hogy a tábla használata során egy hibás törlés történt: egy cellából kitöröltünk egy rekordot a törlés-bit beállítása nélkül. (Vagyis a cellán nem látszik, hogy töröltünk belőle.)
a) Igaz-e, hogy a hibás törlés helye mindig megtalálható?
b) Adjunk hatékony (lineáris időigényű) algoritmust a tábla megjavítására. (Módosítsuk úgy a táblát, hogy megszűnjenek a hibás törlés negatív következményei.)

Gráfelméleti alapfogalmak

1. Legyen $G = (V, E)$ egy irányítatlan (nem feltétlenül egyszerű) gráf, ami éllistával adott. Hogyan lehet $O(|V| + |E|)$ lépésben meghatározni, hogy van-e két azonos fokszámú csúcsa?

2. Egy n pontú és n élű összefüggő gráfnak összesen n feszítőfája van. Mi ez a gráf?

3. Egy gráf izomorf a komplementerével. Mutassuk meg, hogy összefüggő.

4. Hány éle van az n csúcsú teljes gráfnak? És az n csúcsú 4-, illetve 3-reguláris gráfnak? (Egy gráf k-reguláris, ha minden csúcsának foka k.)

5. Bizonyítsd be, hogy egy gráfban a páratlan fokszámú pontok száma páros!

6. Van-e olyan egyszerű gráf, amely csúcsainak fokszáma rendre
a) 1, 2, 2, 3, 3, 3 ?
b) 1, 1, 2, 2, 3, 4, 4 ?
c) 8, 8, 8, 5, 5, 5, 3, 2, 2 ?

7. Egy G egyszerű, n pontú gráfban $(n \geq 3)$ csak egy pontnak van páros foka. Hány páros fokú pont van ekkor G komplementerében? (Egy $G = (V, E)$ gráf komplementere egy $\bar{G} = (V, E')$ gráf, ahol E'-ben pontosan azok az élek szerepelnek, melyek E-ben nincsenek benne.)

8. Egy összefüggő G gráfról tudjuk, hogy minden pontjának foka páratlan, és van egy e éle, melyet elhagyva a gráf két komponensre esik szét. Bizonyítsuk be, hogy ekkor mindkét komponens páratlan sok pontot tartalmaz!

9. Igaz-e, hogy minden összefüggő gráfnak van olyan pontja, melyet (a hozzá tartozó élekkel együtt) elhagyva még mindig összefüggő gráfot kapunk?

10. Van-e olyan (legalább 2 pontú) egyszerű gráf, melyben minden pont foka különböző?

Szélességi bejárás

1. Adott a G irányítatlan gráf a következő éllistával :
$a: b, c; b: a, d; c: a, d; d: b, c, e, f; e: d, f, g; f: d, e, g, h; g: e, f, h; h: f, g;$
Keressünk G-ben a-ból kiinduló szélességi feszítőfát!

2. Legyen G egy irányítatlan összefüggő gráf. Igaz-e, hogy
(a) G minden f éléhez van G-nek olyan szélességi bejárása, amelyben f egy faél?
(b) G minden F feszítőfájához van G-nek olyan szélességi bejárása, amelyben F minden éle faél?

3. Egy játékban egy $n \times m$ rácson lépegetünk. Egy lépés során a rács mentén vízszintesen jobbra vagy függőlegesen lefelé tudunk a következő rácspontba lépni. Azonban adott néhány kereszteződés, ahova nem szabad lépnünk. Adjon $O(nm)$ futási idejű algoritmust annak meghatározására, hogy ha a bal felső rácspontból kezdünk, akkor el tudunk-e jutni a jobb alsó sarokba.

4. Egy $n \times n$-es sakktábla néhány mezőjén az ellenfél egy huszárja (lova) áll. Ha mi olyan mezőre lépünk, ahol az ellenfél le tud ütni, akkor le is üt, de egyébként az ellenfél nem lép. Valamelyik mezőn viszont a mi huszárunk áll. Adjunk $O(n^2)$ lépésszámú algoritmust, ami meghatározza, hogy mely másik mezőkre tudunk (lólépések sorozatával) eljutni a nélkül, hogy az ellenfél leütne!

5. Egy számítógéphálózatban n számítógép van. Minden olyan eseményt, hogy az i-edik gép üzenetet küld a j-ediknek (i, j, t) formában feljegyezünk, ahol a t egész szám az üzenet küldésének időpontját jelöli. Ugyanabban a t időpontban egy gép több gépnek is küldhet üzenetet. Ha a t időpontban az i-edik gép vírusos volt, akkor egy (i, j, t) üzenet hatására a j-edik gép megfertőződhet, ami azt jelenti, hogy a $t+1$ időponttól kezdve már a j-edik gép is vírusos lehet. Legyen adott az (i, j, t) hármasoknak egy m hosszú listája, valamint x, y és $t_0 < t_1$ egész számok. Azt kell eldöntenünk, hogy ha az x-edik gép a t_0 időpontban vírusos volt, akkor lehet-e emiatt az y-adik gép a t_1 időpontban vírusos. Adjon algoritmust, ami ezt a kérdést $O((t_1 - t_0)n + m)$ lépés után megválaszolja.