Algoritmuselmélet zárthelyi

2014. március 31.

1. Legyen $f(x) = \max(x^3 - 10x^2 + 110x; x^2 + 100x)$ és $g(x) = 2^{3\log_2(x)} + x^2$.
   a) Igaz-e, hogy $f = O(g)$? b) Igaz-e, hogy $g = O(f)$?

   Megoldás: Ha x elég nagy, akkor $x^3 - 10x^2 + 110x > x^2 + 100x$. (Könnyű kiszámolni, hogy már $x > 10$-re is.); $g(x) = x^3 + x^2$.
   a) igaz, mert ha $x > 110$, akkor $x^2 > 110x$, ezért $x^3 - 10x^2 + 110x < x^3 - 9x^2 < x^3 + x^2$.
   b) igaz, mert ha $x > 21$, akkor $|g(x)| < 2|f(x)|$, ugyanis ilyenkor $x^3 + x^2 < 2x^3 - 20x^2 + 220x$ nyilván teljesül.

2. Két teherautóval n darab ládát szeretnénk elszállítani. A ládák súlyai $s_1, s_2, \ldots, s_n$ egész számok. Adj olyan algoritmust, ami meghatározza, hogyan kell elhelyezni a ládákat úgy, hogy a két teherautóra rakott összsúly különbsége minimális legyen! Az algoritmus lépésszáma legyen $O(n \cdot w)$, ahol $w = \sum_{i=1}^{n} s_i$.

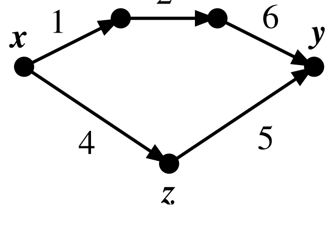
   Megoldás: Alkalmazzuk a hátizsák probléma megoldására tanult dinamikus programozási módszert $v_i = s_i$ értékekkel. Azaz, legyen $P[i,k] = 1$, ha az első i láda közül kiválasztható néhány, aminek a súlyösszege éppen k, különben legyen 0. Ezért $P[0,0] = 1$, valamint $P[i,k] = 0$ ha $i = 0$ és $k > 0$ illetve ha $k < 0$. Az i-edik tárgyat vagy kiválasztjuk, vagy nem, ezért $P[i,k] = \max(P[i-1,k], P[i-1,k-s_i]$. Ezen kívül legyen $B[i,w] = 1$, ha $P[i-1,k-s_i] = 1$ (ez jelenti azt, hogy az i-edik ládát felhasználtuk). Miután kiszámoltunk minden $P[i,k]$ értéket $k \leq w/2$-re, keressük meg a legnagyobb olyan $k \leq w/2$, amire $P[n,k] = 1$. Jelöljük ezt m-el. Ez a legnagyobb összsúly, ami arra a teherautóra kerülhet, ahol a kevesebb összsúly lesz. (Így lesz a különbség minimális.) A $B[i,m]$ érték mondja meg, hogy az i-edik tárgy ezen a teherautón lesz-e.
   Lépésszám: A táblázat egy értékét konstans lépésben kiszámolhatjuk. A táblázat mérete $O(nw)$, utána egy maximum keresés: $O(w)$, összesen $O(nw)$.

3. Legyenek a G gráf pontjai a háromdimenziós tér azon rácspontjai, amelyeknek minden koordinátája 0 és m között van. Két pont pontosan akkor legyen szomszédos, ha egyik koordinátájuk pontosan 1-gyel tér el, a másik két koordinátájuk megegyezik. (Például $(2,3,4)$ és $(2,4,4)$ szomszédosak, de $(2,7,4)$-el nem szomszédos egyik sem.) Mekkora lesz a $(0,0,0)$ pontból indított szélességi keresőfa mélysége?

   Megoldás: A BFS fa mélysége a kiinduló ponttól legtávolabb levő pont távolsága. Ebben a gráfban két tetszőleges pont távolsága a koordinátánként vett különbségek összege (Manhattan távolság). Tehát a fa mélysége $3m$.

4. A Dijkstra és a Bellman-Ford algoritmus is úgy működik, hogy amikor meghatározza az adott pontba mutató legrövidebb út hosszát, akkor valójában felfedezett egy ekkora hosszúságú legrövidebb utat. Adj példát olyan irányított gráfra, melyben minden élen különböző, egész, pozitív súlyok vannak, és a Dijkstra illetve Bellman-Ford két különböző legrövidebb utat talál az x pontból az y pontba!

   Megoldás: Például a következő gráf:



Ekkor a legrövidebb út nyilván 9. A Dijkstra először megtalálja a v-be vezető 1 hosszú utat, és a z-be vezető 4 hosszút. A következő körben v-ből próbálunk továbblépni, megtaláljuk az xvw utat, mint legrövidebb út, majd a harmadik körben az $xvwy$ utat. (Az xzy út csak ez után kerül megvizsgálásra, de mivel nem rövidebb, mint a másik, nem változtatunk.)
A Bellman-Ford az első körben megtalálja az egy élből álló utakat v-be és z-be. A második körben a két élből állókat y-ba és w-be, tehát ilyenkor előbb az xzy-t találja meg.

5. Az $A[1:2n]$ tömb egy kupacot reprezentál.
   a) Igaz-e, hogy az $A[1:n]$ tömb biztosan egy kupacot reprezentál?
   b) Igaz-e, hogy az $A[n+1:2n]$ tömb biztosan egy kupacot reprezentál?

   Megoldás: Ha kupac, akkor $A[i] < A[2i]$ és $A[i] < A[2i+1]$.
   a) Az előbbi tulajdonság nyilván teljesül akkor is, ha csak az első n elemet nézzük, tehát kupac.
   b) Nem biztos, ellenpélda: $1, 2, 3, 6, 5, 4$. Ekkor $6, 5, 4$ nem kupac.

6. A $B[1:n]$ tömb különböző egészeket tartalmaz. A $B[i]$ elem *lokális minimum*, ha $B[i-1] > B[i]$ és $B[i] < B[i+1]$ teljesül ($B[1]$ ill. $B[n]$ elég, ha az egy szomszédjánál kisebb). Adj algoritmust egy lokális minimum megkeresésére, mely legrosszabb esetben $O(\log n)$ összehasonlítást használ! (Ha több lokális minimum van, ezek közül mindegy melyiket találjuk meg.)

   Megoldás: Hasonlítsuk össze elsőnek $B[\lfloor n/2 \rfloor]$-t és $B[\lfloor n/2 \rfloor + 1]$-et. Ha $B[\lfloor n/2 \rfloor] < B[\lfloor n/2 \rfloor + 1]$, akkor belátható, hogy a $B[1 : \lfloor n/2 \rfloor]$-ben van egy lokális minimum. Ugyanis, lépegessünk egyesével balra $B[\lfloor n/2 \rfloor]$-tól, amíg mindig kisebb elemet találunk. Ha a következő elem már nagyobb, akkor találtunk egy lokális minimumot. Ha pedig elérünk $B[1]$-hez, akkor $B[1]$ a lokális minimum. Ha $B[\lfloor n/2 \rfloor] > B[\lfloor n/2 \rfloor + 1]$, akkor hasonlóan belátható, hogy a $B[\lfloor n/2 \rfloor + 1 : n]$-ben van egy lokális minimum. Így visszavezettük a feladatot egy fele méretűre. Ez folytatva $O(\log n)$ összehasonlítással megtalálunk egy lokális minimumot, hiszen $O(\log n)$-szer kell felezni az intervallumot, hogy 1 hosszú legyen.

7. Az $1, 2, 3, 4, 5, 8, 10, 9, 7, 6$ tömböt gyorsrendezéssel rendezzük, amit kétféleképpen is végrehajtunk. Az egyik futtatás során mindig az éppen rendezendő tömb első elemét választjuk véletlen elemnek a partíciós lépéshez, a másik futtatás során mindig a tömb utolsó elemét. Melyik esetben fogunk kevesebb összehasonlítást végezni?

   Megoldás: Amikor egy k-elemű tömbre futtatjuk a partíciós eljárást, akkor $k-1$ összehasonlítást végzünk. Amikor az első elemet választjuk a partícióhoz, akkor az első 5 partíció során az összehasonlítások száma $9 + 8 + 7 + 6 + 5 = 35$, ennyi összehasonlítást tehát biztos végzünk. (Kiszámolható, hogy 41 összehasonlítás lesz.)
   Amikor az utolsó elemet választjuk, akkor az első partíció 9 összehasonlítást végez és ezzel egy 5 és egy 4 méretű tömb rendezésére vezeti vissza a feladatot. Ezeket a gyorsrendezés legrosszabb esetben is 10 illetve 6 összehasonlítással rendezi, vagyis ilyenkor legfeljebb $9 + 10 + 6 = 25$ összehasonlítást használ (pontosan 24 lesz), ami kevesebb, mint a másik esetben.

8. Egy piros-fekete fában 13 elemet tárolunk. Minimálisan hány piros csúcs van a fában? (A teljes megoldáshoz be kell látni egy megfelelő k-ra, hogy lehet k piros, és azt is, hogy nem lehet $k-1$ piros.)

   Megoldás: Belátjuk, hogy legkevesebb 2 piros csúcs van. Ha nincs piros csúcs egy fában, akkor mivel a gyökértől minden levél egyforma távolságra van, ezért $2^k - 1$ elemet tudunk tárolni valamilyen k-ra. Mivel ez semmilyen k-ra nem 13, ez nem lehet. Ha pontosan 1 piros van (ami nem lehet a gyökér), akkor legyen ennek fekete-magassága i. Ennek a pontnak minden leszármazottja fekete, ezért összesen $2(2^i - 1)$ leszármazottja van. Ez a pirossal együtt 2^i-vel több, mintha a piros helyett egy fekete gyökerű ugyanolyan mélységű csupa fekete fa lenne. Tehát ilyenkor a fában $2^k + 2^i - 1$ elemet tárolunk, ami nem lehet 13. Két piros csúcs viszont lehet: