Algoritmusok és gráfok
MÁSODIK GYAKORLAT, 2019. szeptember 20.
Megoldások néhány kiválasztott feladathoz

1. (a) Az alábbi pszeudokód egy $n \geq 2$ méretű tömbben a tömb végére mozgatja a legnagyobb elemet. Erre a kódra alapozva írja le pszeudokóddal az alábbi, buborékrendezés nevű eljárást: Az első fázisban lefuttajuk ezt a kódot a teljes $A[0:n-1]$ tömbre, a második fázisban az így kapott tömb $A[0:n-2]$ résztömbjére, a harmadik fázisban az így kapott tömb $A[0:n-3]$ résztömbjére, stb., végül az $(n-1)$. fázisban az előző körben kapott tömb $A[0:1]$ résztömbjére.

```
ciklus i = 0-tól (n-2)-ig:
    ha   A[i] > A[i+1]:
        csere A[i] és A[i+1]
ciklus vége
```

(b) Lássa be, hogy az a) részben adott pszeudokód rendezi a tömböt.
(c) Mutassa meg, hogy a buborékrendezés lépésszáma $O(n^2)$ (lépésnek az összehasonlítás és a csere számít).

Megoldás

(a) Hasonlóan a kiválasztásos rendezéshez, itt is két, egymásba ágyazott ciklusra lesz szükségünk a pszeudokódban. A külső ciklus fogja szabályozni azt, hogy mekkora annak a résztömbnek a hossza, amiben dolgozunk (amin belül a legnagyobb elemet a végére mozgatjuk), a belső ciklus pedig azt fogja megvalósítani, hogy a külső ciklus által definiált résztömbön végigmenve a legnagyobb elemet a résztömb végére mozgatja.

```
ciklus j = n-1-től 1-ig:   // az A[0:j] tömbben dolgozunk
    ciklus i = 0-tól (j-1)-ig:
        ha A[i]> A[i+1]:
            csere A[i] és A[i+1]
    ciklus vége
ciklus vége
```

A fenti kódban a // jel utáni rész a komment, amit azért írunk ide, hogy a j ciklusváltozó szerepét elmagyarázzuk.
(b) Ez az eljárás helyes, mert a külső ciklus magjának első futása után az input tömb legnagyobb eleme a tömb végére kerül, vagyis éppen oda, ahol a rendezett sorban állnia kell és később innen soha el nem mozdítjuk. A ciklusmag 2. futása után a maradék elemek közül a legnagyobb, vagyis a tömb 2. legnagyobb eleme az utolsó előtti cellába kerül és innen soha el nem mozdítjuk később. Általában is igaz, hogy a k. futás után a tömb k legnagyobb eleme már a helyén lesz és soha nem mozdul el onnan később, így az eljárás végére mindenki a helyére fog kerülni.
(c) A külső ciklus legfeljebb n-szer fut le (igazából $n-1$-szer fut le pontosan) és minden lefutása $O(n)$ lépés, mert egy legfeljebb n hosszú tömbben futattjuk azt az algoritmust, aminek lépésszáma ℓ méretű tömb esetén $O(\ell)$. Ez azt jelenti, hogy (az előadáson látott "számolási szabály" alapján, miszerint $n \cdot O(n)$ az $O(n^2)$) a buborékrendezés $O(n^2)$-es algoritmus.

2. Lássa be, hogy az első (múlt órai) feladatsor 6. feladatában leírt eljárás lépésszáma $O(n)$. (Lépésnek az értékadás és az összeadás számít.)

Megoldás

Értékadásból van 2 az elején és pontosan $n-2$ a cikluson belül, összeadásból pedig $n-2$ van, vagyis a lépésszám $T(n) = 2 + 2(n-2) = 2n-2$. Mivel $T(n) = 2n-2 \leq 2n$ fennáll, ha $n \geq 1$, így teljesül a O definíció $c = 2$, $n_0 = 1$ és $f(n) = n$ esetén, vagyis $T(n)$ $O(n)$-es.

3. Igaz-e, hogy egy algoritmus lépésszáma $O(n^2)$, ha tudjuk, hogy a lépésszám
(a) $10n^2 - n \log n$, (b) $n + n^2 + n^3$, (c) $10000 \log \log n$

Megoldás

(a) Igaz, mert $T(n) = 10n^2 - n \log n \leq 10n^2$ minden $n \geq 1$ esetén igaz, így $c = 10$, $n_0 = 1$ jó választás

(b) Nem igaz, mert ha $T(n) = n + n^2 + n^3 \leq c \cdot n^2$ fennállna valami c konstanssal valami n_0 küszöbértéktől kezdve, akkor $n^3 \leq n + n^2 + n^3 \leq c \cdot n^2$ is igaz lenne, vagyis $n^3 \leq c \cdot n^2$ igaz lenne, ha $n \geq n_0$. Ezt az egyenlőtlenséget elosztva n^2-tel azt kapjuk, hogy $n \leq c$, ha $n \geq n_0$, ami nem igaz, vagyis a kezdeti feltevésünk ben volt a hiba, miszerint $T(n)$ $O(n^2)$-es.

(c) Igaz, mert $10000 \log \log n \leq 10000n^2$, ha $n \geq 1$ vagyis $c = 10000$ és $n_0 = 1$ jó választás.

4. **(ZH 2018)** Igaz-e, hogy ha egy algoritmus lépésszáma $100 \cdot n^2 + 10^{10} \cdot n + 17$, akkor az algoritmus lépésszáma $O(n^2)$? Ha úgy véli, hogy ez igaz, akkor megfelelő c konstans és n_0 küszöbérték megadásával lássa ezt be, ha pedig úgy véli, hogy hamis, akkor bizonyítsa be ezt.

Megoldás

Az állítás igaz. Ezt úgy fogjuk megmutatni, hogy adunk olyan c konstanst és n_0 küszöbértéket, melyekre $100 \cdot n^2 + 10^{10} \cdot n + 17 \leq c \cdot n^2$ teljesül, ha $n \geq n_0$.

Egy lehetséges jó becslés a következő:

$$100 \cdot n^2 + 10^{10} \cdot n + 17 \leq 100 \cdot n^2 + 10^{10} \cdot n^2 + 17n^2 = (100 + 10^{10} + 17)n^2$$

ami igaz, ha $n \geq 1$ és ahol csak azt használtuk, hogy $n \leq n^2$.

Így a $c = 100 + 10^{10} + 17$ és $n_0 = 1$ választás megfelelő.

Egy másik lehetőség lett volna a

$$100 \cdot n^2 + 10^{10} \cdot n + 17 \leq 100 \cdot n^2 + n^2 + n^2 = 102n^2$$

becslés használata, ami akkor igaz, ha $n \geq 10^{10}$, vagyis ebben az esetben a $c = 102$, $n_0 = 10^{10}$ értékekkel teljesül a $O(n^2)$ definíciója.

5. **(Mintavizsga 2018)** Az alábbi pszeudokód inputja két, egész számokat tartalmazó n méretű tömb, A és B. Mutassa meg, hogy a pszeudokód által leírt algoritmus lépésszáma $O(n^2)$. (Egy lépésnek számít egy számról eldönteni, hogy páros-e, egy lépés az értékadás és két szám összedása.)

```
ciklus i = 0-tól  (n-1)-ig:
    ha A[i] páros:
        ciklus  j = 0-tól  (n-1)-ig:
            B[j] : = B[j] + 17
        ciklus vége
ciklus vége
```

Megoldás

A külső ciklus n-szer fut le. Megmutatjuk, hogy a ciklusmag minden egyes lefutása $O(n)$ lépésszámú, így az összes lépésszám $n \cdot O(n)$-nel becsülhető, ami a tanultak szerint $O(n^2)$.

A ciklusmagban egy lépés a párosság eldöntése, majd (ennek eredményétől függően) legfeljebb n összeadás és n értékadás van, azaz a ciklusmak legfeljebb $1 + 2n$ lépésszámú. Mivel $1 + 2n \leq 3n$, ha $n \geq 1$, ezért a ciklusmag $O(n)$-es ($c = 3$, $n_0 = 1$).