Algel I. gyakorlat, 2015. február. 11.
Barátkozás a tárggyal és egymással

Drótos Márton
drotos@cs.bme.hu

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

2. Bizonyítsuk be, hogy

   (a) $\log_2 f(n) = \Theta(\log_{100} f(n)) \quad (f(n) > 0)$,

   (b) $f(x) = a_k x^k + a_{k-1}x^{k-1} + \cdots + a_0 \quad (a_k \neq 0) \quad \Rightarrow \quad f(n) = \Theta(n^k)$,

   (c) $2^{n+1} = O(2^n)$, de $2^{2n} \neq O(2^n)$,

   (d) $\max(f(n), g(n)) = \Theta(f(n) + g(n)) \quad (f(n), g(n) > 0)$!

3. **[Vizsga: 2007. június 12.]** Egy $\mathcal{A}$ algoritmusról azt tudjuk, hogy 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 bemenet hosszát jelöli.)

4. **[ZH: 2010. április 19.]** Legyen $f_1(n) = n^{3 \log n}$ és $f_2(n) = 2010 \cdot 4^{\log n \cdot \log n}$. Igaz-e, hogy $f_1 = O(f_2)$, illetve, hogy $f_2 = O(f_1)$?

5. Jelöljük $T(n)$-nel egy algoritmus legnagyobb lehetséges lépésszámát az n méretű inputokon. Tudjuk, hogy $T(n) \leq 10$, ha $n \leq 5$ és $T(n) \leq T(n-1) + n/3$, ha $n > 5$. Ekkor mit tudunk mondani $T(n) = O(n)$, $T(n) = O(n^2)$ és $T(n) = O(n^3)$ egyenlőségek helyességéről?

6. **[ZH: 2011. március 28.]** Egy problémára két algoritmusunk van.

   Az $\mathcal{A}$ algoritmus az $n \geq 2$ méretű problémából 10 lépéssel 2 db $n-1$ méretűt készít és ezeket oldja meg rekurzívan.

   A $\mathcal{B}$ az $n \geq 2$ problémából 3 lépéssel 4 db $n-1$ méretűt készít és ezeket oldja meg rekurzívan. Az $n = 1$ esetben mindkét eljárás 1 lépést használ.

   Melyik algoritmus lesz nagy n értékekre gyorsabb?

---

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 alábbi függvényeket rendezzük olyan sorozatba, hogy ha f_i után közvetlenül f_j következik a sorban, akkor $f_i(n) = O(f_j(n))$ teljesüljön!

$$f_1(n) = 8n^{2.5}, \quad f_2(n) = 5\sqrt{n} + 1000n, \quad f_3(n) = 2^{\log^2 n}, \quad f_4(n) = 2008n^2 \log n$$

9. Az $\mathcal{A}$ algoritmusról azt tudjuk, hogy n hosszú bemeneteken a lépésszáma $O(n^2)$. Lehetséges-e, hogy

   (a) $\forall n$ hosszú bemeneten $O(n)$ lépést használ?

   (b) $\exists x$, hogy az x bemeneten az algoritmus lépésszáma $10|x|^2 \log |x| - 800$ (ahol $|x|$ az x bemenet hosszát jelöli)?

10. Adjunk O becslést a következő függvényekre:

(a) $(n^2+8)(n+1)$
(b) $(n\log_2 n+n^2)(n^3+2)$
(c) $(n^3+n^2\log_2 n)(\log_2 n+1)+(17\log_2 n+19)(n^3+2)$
(d) $(n!+2^n)(n^3+\log_2(n^2+1))$
(e) $(2^n+n^2)(n^3+3^n)$
(f) $(n^n+n2^n+5^n)(n!+5^n)$

11. **[Vizsga: 2007. június 19.]** Az alábbi függvényeket rendezze olyan sorozatba, hogy ha f_i után közvetlenül f_j következik a sorban, akkor $f_i(n)=O(f_j(n))$ teljesüljön!

$$f_1(n)=2^{100n}-2^{50n},\quad f_2(n)=2007n^3,\quad f_3(n)=3^{3n}$$

12. **[Vizsga: 2007. június 5.]** Jelölje egy algoritmus maximális lépésszámát az n hosszú bemeneteken $L(n)$. Azt tudjuk, hogy minden $n=2k>4$ páros számra $L(2k)\le L(2k-2)+1$ teljesül, és hogy $L(4)=10$. Következik-e ebből, hogy az algoritmus lépésszáma $O(n)$?

13. **[PZH: 2011. április 22.]** Egy problémára két algoritmusunk van.

    Az $\mathcal{A}$ algoritmus az $n\ge 2$ méretű problémából 5 lépéssel 2 db legfeljebb $n/2$ méretűt készít és ezeket oldja meg rekurzívan.

    A $\mathcal{B}$ algoritmusról azt tudjuk, hogy lépésszáma az n méretű problémákon $O(n^2)$.

    Ha ennyiből lehetséges, határozza meg, melyik algoritmus lesz nagy n értékekre gyorsabb! Ha ennyi információból még nem következik, hogy $\mathcal{A}$ vagy $\mathcal{B}$ lesz a gyorsabb, akkor indokolja meg, miért nem!

14. **[PZH: 2012. április 25.]** Tudjuk, hogy az $f(n),g(n):\mathbb{N}\to\mathbb{N}$ függvényekre igaz, hogy $f(n)=O(g(n))$, de $f(n)\ne\Theta(g(n))$. Lehetséges-e, hogy $f(n)+2012\cdot n^{\log n}=\Theta(n\log n+g(n))$?

15. Tudjuk, hogy $f(x)=O(h(x))$ és $g(x)=O(h(x))$. Igaz-e, hogy

    (a) ha $h(x)=3x$, akkor $f(g(x))=O(h(x))$;
    (b) $f(g(x))=O(h(x))$ $\forall h$ függvényre?