Függvények nagyságrendje; Keresés

Adatstruktúrák és algoritmusok
2. gyakorlat

2015. február 13.

Def. Ha $f(n)$ és $g(n)$ a természetes számokon értelmezett, pozitív értékeket felvevő függvények, akkor

i. $f = O(g)$ jelöli azt a tényt, hogy vannak olyan $c, n_0 > 0$ állandók, hogy $n \geq n_0$ esetén $f(n) \leq c \cdot g(n)$ teljesül.

ii. $f = \Omega(g)$ jelöli azt a tényt, hogy vannak olyan $c, n_0 > 0$ állandók, hogy $n \geq n_0$ esetén $f(n) \geq c \cdot g(n)$ teljesül.

iii. $f = \Theta(g)$ jelöli azt a tényt, hogy $f = O(g)$ és $f = \Omega(g)$ is teljesül.

1. Bizonyítsuk be, hogy

   (a) $2x^2 + x - 1 = \Theta(x^2)$;

   (b) $x^2 + 4x + 17 = O(x^3)$, de $x^3 \neq O(x^2 + 4x + 17)$;

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

   (d) $1 + 2 + \ldots + n = O(n^2)$, de $1 + 2 + \ldots + n \neq O(n)$.

2. 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.

3. Tekintsük az $f_1(n) = 2009n!$ és $f_2(n) = 100(n-1)!$ függvényeket. Igaz-e, hogy
   a) $f_1 = O(f_2)$ b) $f_2 = O(f_1)$ c) $f_1 = \Omega(f_2)$ d) $f_2 = \Omega(f_1)$?

4. Adjunk O becslést a következő függvényekre.

   (a) $(n^2 + 8)(n + 1)$

   (b) $(n \log_2 n + n^2)(n^3 + 2)$

5. Melyik az a legkisebb egész k, amelyre igaz, hogy az alábbi kifejezés $O(x^k)$?

   (a) $2x^2 + x^3 \log x$

   (b) $(x^4 + x^2 + 1)/(x^4 + 1)$

(c) $(x^3 + 5\log x)/(x^4 + 1)$

6. Mi lehet $T(n)$, ha $T(1) = 2$ és $n \geq 2$ esetén $T(n) = 3 \cdot T(n-1) + 1$?

7. Egy problémára két algoritmusunk van.

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

   A $\mathcal{B}$ algoritmus az $n \geq 2$ méretű problémából 3 lépéssel 4 darab $n-1$ méretűt készít és ezeket oldja meg rekurzívan.

   Az $n = 1$ esetben mindkét eljárás 1 lépést használ.

   Melyik algoritmus lesz nagy n értékekre gyorsabb?

8. 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,

   (b) n^3,

   (c) 2^n?

---

1. Gondoltam egy számot 0 és 31 között. Nyilván ki lehet barkochbázni 5 kérdéssel. Adjunk meg előre 5 kérdést úgy, hogy az azokra adott válaszokból kitalálható legyen a gondolt szám!

2. Adott az $A[1:n]$ csupa különböző egész számot növekvő sorrendben tartalmazó tömb. 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!

3. Pontosan hány összehasonlítás kell ahhoz, hogy egy n darab szám közül a legkisebb 10 szám egyikét megtaláljuk? (Az eredmény bármelyik lehet a legkisebb tíz közül: tehát pl. az első éppúgy megfelel, mint a tizedik.)