

# Nagyságrendek

Kiegészítő anyag az Algoritmuselmélet tárgyhoz  
(a Rónyai–Ivanyos–Szabó: Algoritmusok könyv mellé)

Friedl Katalin  
BME SZIT  
friedl@cs.bme.hu

2018. február 1.

## Az $O$ , $\Omega$ , $\Theta$ jelölések

Az algoritmusok elemzése során a lépésszámot a bemenet hosszának függvényében szokás meghatározni. Ebbe a lépésszámba általában csak a „fontos” lépéseket számoljuk bele, például hogy egy rendezési feladat során hány összehasonlítást végzünk, a „nem fontos” lépéseket, pl. egy ciklusváltozó növelését nem. A lépésszám pontos meghatározása helyett általában elegendő a lépésszám nagyságrendjének meghatározása, ebből már (kis óvatossággal) lehet következtetni arra, hogy az algoritmus mennyire hatékony.

Egy másik szempont, ami miatt a nagyságrendek hasznosak, hogy ezek alapján elég jól meg tudjuk jósolni, hogy ha nagyobb bemenetre akarjuk az algoritmusunkat használni, akkor mennyivel fog tovább dolgozni. Például tegyük fel, hogy egy program az  $n_0 = 20$  méretű teszteseteken 6 másodpercig fut. Meddig fog futni, egy  $n_1 = 400 = 20n_0$  méretű bemeneten? Jelölje  $f(n)$  az algoritmus maximális lépésszámát az  $n$  hosszú bemeneteken. Nézzünk néhány esetet:

**Lineáris függvény:**  $f(n) = cn$

A futási idő 20-szorosára nő, azaz 2 percig is eltart.

**Másodfokú függvény:**  $f(n) = cn^2$

A futási idő 20<sup>2</sup>-szeresére nő, azaz 40 percig is eltarthat.

**Logaritmikus függvény:**  $f(n) = c \log n$

Mivel  $n_1 = n_0^2$ , a futási idő 2-szeresére nő, azaz csak 12 másodperc lesz.

**Exponenciális függvény:**  $f(n) = c2^n$

A futási idő erre  $6 \cdot 2^{380}$  másodperc lesz, ami felhasználva, hogy  $2^{10} > 10^3$ , több mint  $6 \cdot 10^{114}$  másodperc. Mennyi is ez az idő? Mivel egy év kb.  $3 \cdot 10^7$  másodperc, ez kb.  $2 \cdot 10^{107}$  év lenne, ami még akkor is nagyon sok, ha

párhuzamos számítógépünk van: a világegyetem állítólag nem több, mint  $10^{80}$  részecskéből áll, tehát abban az esetben is eltartana a számítás  $2 \cdot 10^{27}$  évig, ha minden részecske nekünk dolgozna. Ezt azzal érdemes összevetni, hogy jelenleg a Föld korát  $10^{10}$  évnél kevesebbre becsülik. Látszik, hogy jelentősen gyorsabb processzorokkal is reménytelen a program ekkora bemenetre való lefuttatása.

Ha kicsit összetettebb a függvény akkor is hasonló becslés adható, pl.  $cn^2+1$  alakú függvénynél is ha  $n$   $\alpha$ -szorosára nő, a függvény értéke kb.  $\alpha^2$ -szeresére. Ennek pontosabb megfogalmazását segítik a következő fogalmak.

Legyen  $f$  és  $g$  két, a természetes számokon értelmezett valós függvény.

**1. Definíció.** Az  $f \in O(g)$  jelölés azt jelenti, hogy van olyan  $c > 0$  valós konstans és  $n_0$  pozitív egész küszöbszám, hogy minden  $n \geq n_0$  esetén  $|f(n)| \leq c|g(n)|$  teljesül.

**1. Megjegyzés.** Szóban ez úgy hangzik, hogy az  $f$  függvény nagy ordó  $g$ , vagy röviden: ordó  $g$ .

Egy rögzített  $g$  függvényre az  $O(g)$  tekinthető a függvények azon halmazának, melyekre  $f \in O(g)$ , ez indokolja az  $\in$  jelölést.

**2. Definíció.** Az  $f \in \Omega(g)$  jelölés azt jelenti, hogy van olyan  $c > 0$  valós konstans és  $n_0$  pozitív egész küszöbszám, hogy minden  $n \geq n_0$  esetén  $|f(n)| \geq c|g(n)|$  teljesül.

**2. Megjegyzés.** Szóban ez úgy hangzik, hogy az  $f$  függvény nagy omega  $g$ , vagy röviden: omega  $g$ .

Néhány egyszerű következmény:

- Ha  $g(n) \leq h(n)$  teljesül minden  $n$ -re, akkor  $O(g) \subseteq O(h)$ , és  $\Omega(g) \supseteq \Omega(h)$ , hiszen ha  $f \in O(g)$ , akkor  $f \in O(h)$  is (Miért?) és ha  $t \in \Omega(h)$ , akkor  $t \in \Omega(g)$  (Miért?).
- Ha  $f \in O(g)$ , akkor  $O(f) \subseteq O(g)$ , hiszen ha  $|h(n)| \leq c_1|f(n)|$  teljesül minden  $n \geq n_1$  esetén és  $|f(n)| \leq c_2|g(n)|$  teljesül minden  $n \geq n_2$  esetén, akkor  $|h(n)| \leq c_1c_2|g(n)|$ , ha  $n \geq \max(n_1, n_2)$ .
- Ha  $f \in O(g)$ , akkor  $g \in \Omega(f)$ , hiszen ha  $|f(n)| \leq c|g(n)|$ , akkor  $\frac{1}{c}|f(n)| \leq |g(n)|$ .

**3. Definíció.** Az  $f \in \Theta(g)$  jelölés azt jelenti, hogy  $f \in O(g)$  és  $f \in \Omega(g)$  egyaránt teljesül.

**3. Megjegyzés.** Szóban ez úgy hangzik, hogy az  $f$  függvény nagy teta  $g$ , vagy röviden: teta  $g$ .

A definíciókból látszik, hogy  $f \in \Theta(g)$  pontosan akkor teljesül, ha van olyan  $c_1, c_2 > 0$  valós konstans és  $n_0$  pozitív egész küszöbszám, hogy minden  $n \geq n_0$  esetén  $c_1|g(n)| \geq |f(n)| \geq c_2|g(n)|$  teljesül.

Általában úgy használjuk ezeket, hogy a definíciókban levő  $g$  valami egyszerű függvény (pl.  $n, n^2, \log n, 2^n$ ). Az  $O$  egy felső becslést, az  $\Omega$  egy alsó becslést ír le a függvény nagyságrendjére, a  $\Theta$  a pontos nagyságrend megadására alkalmas.

**4. Megjegyzés.** Szokásos az  $f = O(g)$ ,  $f = \Omega(g)$ , illetve  $f = \Theta(g)$  jelölés is.

## Példák

1. Legyen  $f(n) = 3n + 1000$  és  $g(n) = n$ .
  - (a) Ekkor  $f \in O(g)$ , mert például  $c = 4$ ,  $n_0 = 1000$  választással  $|f(n)| = f(n) = 3n + 1000 \leq 4n = cn$  ha  $n \geq n_0 = 1000$ .  
(Más jó választás is van, minden  $c > 3$  értékhez találhatunk megfelelő  $n_0$  értéket, de az állítás igazolásához elegendő egyetlen megfelelő  $c$ ,  $n_0$  párt megadni.)
  - (b)  $f \in \Omega(g)$  is teljesül, pl. a  $c = 3$ ,  $n_0 = 1$  választással  $|f(n)| = f(n) = 3n + 1000 \geq 3n = cn$  ha  $n \geq n_0$ .  
(Más jó választás is van, minden  $c \leq 3$  érték megfelelő.)
  - (c)  $f \in \Theta(g)$  igaz, mert az előbb láttuk,  $f \in O(g)$  és  $f \in \Omega(g)$ .
2. Legyen  $f(n) = 3n + 1000$  és  $g(n) = n^2$ .
  - (a) Ekkor  $f \in O(g)$ , mert  $|f(n)| = f(n) = 3n + 1000 \leq 4n \leq n^2$  igaz, ha  $n \geq 1000$  (azaz  $c = 1$ ,  $n_0 = 1000$  jó).  
Vagy mondhatjuk azt is, hogy az előző példában láttuk, hogy  $f \in O(n) \subseteq O(n^2)$ , tehát valóban  $f \in O(g)$ .
  - (b)  $f \notin \Omega(g)$  mert  $|f(n)| = f(n) = 3n + 1000 \leq 4n$ , és ezért ha  $|f(n)| \geq cn^2$  teljesül, akkor  $4n \geq cn^2$  is igaz kell legyen. Ez pedig, akárhogy is választjuk a  $c > 0$  konstans, nagy  $n$ -ekre ( $n > 4/c$ ) nem teljesül.
  - (c)  $f \notin \Theta(g)$ , mert az előző pont szerint  $f \notin \Omega(g)$ .
3. Legyen  $f(n) = 3n^2 - 100n + 6$ .
  - (a)  $f(n) \in O(n^2)$ . Vegyük észre, ha  $n$  elég nagy, akkor  $f(n) > 0$ , így feltehetjük, hogy  $|f(n)| = f(n)$ . Például ez biztos igaz, ha  $3n^2 \geq 100n$ , tehát  $n \geq 34$ . (A megoldóképlettel is ki lehet számolni, honnantól nem negatív  $f(n)$  értéke, de most nekünk elég egy érvényes becslést adni.) Tehát nézzük csak az  $n \geq 34$  helyeket. Ekkor  $|f(n)| = f(n) = 3n^2 - 100n + 6 \leq 3n^2$ , mivel  $100n \geq 6$ . Azaz a  $c = 3$ ,  $n_0 = 34$  jó választás.
  - (b)  $f(n) \in O(n^3)$ , mert az előzőhöz hasonlóan foglalkozhatunk csak az  $n \geq 34$  esettel, amikor  $|f(n)| = f(n) \leq 3n^2 \leq n^3$ , tehát a  $c = 1$ ,  $n_0 = 34$  jó választás.

Vagy: mivel láttuk már, hogy  $f \in O(n^2) \subseteq O(n^3)$ , ezért  $f(n) \in O(n^3)$ .

- (c)  $f(n) \notin O(n)$ , mert, ismét csak az  $n \geq 34$  esetet tekintve láthatjuk, hogy  $|f(n)| = f(n) = 3n^2 - 100n + 6 \geq 2n^2$ , feltéve, hogy  $n^2 \geq 100n$ , azaz  $n \geq 100$  is teljesül. Mivel  $2n^2 > cn$  minden  $c$ -re, ha  $n > c/2$ , azt kapjuk, hogy  $f(n) \not\leq cn$ , ha  $n$  elég nagy (pontosabban, ha  $n > \max\{100, c/2\}$ ).
- (d)  $f(n) \in \Omega(n^2)$ , hiszen az előbb láttuk, hogy  $f(n) \geq 2n^2$ , ha  $n \geq 100$ .
- (e)  $f(n) \in \Theta(n^2)$ , mert  $f(n) \in O(n^2)$  és  $f(n) \in \Omega(n^2)$  is teljesül.
4.  $\log_a n \in \Theta(\log_2 n)$ , hiszen  $\log_a n = (\log_2 n)/(\log_2 a)$ , így tehát  $c_1 = c_2 = 1/(\log_2 a)$  választással minden  $n > \max\{a, 2\}$  számra teljesül, hogy  $c_1 \log_2 n = \log_a n = c_2 \log_2 n$ .
5.  $2^{2n} \notin O(2^n)$ , hiszen  $2^{2n} = 2^n \cdot 2^n$ , tehát nagy  $n$ -ekre semmilyen  $c$  esetén sem teljesül, hogy  $2^{2n} \leq c2^n$  (csak ha  $2^n \leq c$ , azaz  $n \leq \log c$ ).
6. Megmutatható, hogy ha  $f(n) = a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0$  és  $a_k \neq 0$ , akkor  $f(n) \in \Theta(n^k)$ .
7.  $\max(f(n), g(n)) \in \Theta(f(n) + g(n))$ , feltéve, hogy  $f(n), g(n) > 0$  minden  $n$ -re. Ez azért igaz, mert ha mindkét érték pozitív, akkor egyrészt  $f(n)$  és  $g(n)$  is kisebb mint  $f(n) + g(n)$ , tehát  $\max(f(n), g(n)) < f(n) + g(n)$ , másrészt  $\max(f(n), g(n)) \geq \frac{f(n)+g(n)}{2}$ .

A leggyakrabban előforduló nagyságrendekre érvényesek a következő tartalmazások (Miért?)

$$O(\log n) \subset O(n) \subset O(n^2) \subset O(n^3) \subset O(2^n) \subset O(3^n) \subset O(n!) \subset O(n^n) \subset O(2^{2^n})$$

$$\Omega(\log n) \supset \Omega(n) \supset \Omega(n^2) \supset \Omega(n^3) \supset \Omega(2^n) \supset \Omega(3^n) \supset \Omega(n!) \supset \Omega(n^n) \supset \Omega(2^{2^n})$$

## Feladatok

1. Az alábbi függvényeket rendezze olyan sorozatba, hogy ha  $f_i$ -t  $f_j$  követi a sorban, akkor  $f_i(n) \in O(f_j(n))$  teljesüljön!

$$f_1(n) = 11n^{2,5}, \quad f_2(n) = 5\sqrt{n} + 1000n,$$

$$f_3(n) = 2^{\log^2 n}, \quad f_4(n) = 2007n^2 \log n.$$

*Megoldás:*  $f_2(n) = 5\sqrt{n} + 1000n \leq 1005n$ , ha  $n \geq 1$  és ha  $n > 2$ , akkor  $1005n \leq 2007n^2 \log n$ , ezért  $f_2 \in O(f_4)$ .

Mivel  $\log n \leq \sqrt{n}$ , ezért  $2007n^2 \log n \leq 2007n^2 \sqrt{n} = \frac{2007}{11}(11n^{2,5})$ , tehát  $f_4 \in O(f_1)$ .

Írjuk át  $f_3$ -at:  $2^{\log^2 n} = 2^{\log n \cdot \log n} = (2^{\log n})^{\log n} = n^{\log n}$ . Ha  $n > 6$ , akkor  $\log n > 2,5$ , és ezért ilyenkor  $f_1(n) = 11n^{2,5} \leq 11n^{\log n} = 11f_3(n)$ . Tehát az  $f_2, f_4, f_1, f_3$  sorrend jó.

(Meggondolható, hogy ez az egyetlen helyes sorrend.)

2. Az  $\mathcal{A}$  algoritmról tudjuk, hogy  $n$  hosszú bemeneteken a lépésszáma  $O(n^2)$ .
- (a) Lehetséges-e, hogy minden  $n$  hosszú bemeneten  $O(n)$  lépést használ?
  - (b) Következik-e a feltételből, hogy minden  $n$  hosszú bemeneten  $O(n)$  lépést használ?
  - (c) Lehetséges-e, hogy van olyan  $x$ , hogy az  $x$  bemeneten az algoritmus lépésszáma  $10|x|^2 \log |x| - 800$  (ahol  $|x|$  az  $x$  bemenet hosszát jelöli)?

*Megoldás:* Jelölje  $f(n)$  az algoritmus maximális lépésszámát az  $n$  hosszú bemeneteken. A feltétel szerint  $f(n) \in O(n^2)$ , azaz elég nagy  $n$  értékekre  $f(n) \leq cn^2$  teljesül valamilyen pozitív  $c$ -re.

Az (a)-ra a válasz igen, hiszen a feltétel csak egy felső becslés, lehet pl. hogy minden  $n$ -re  $f(n) = n$ .

(b) Nem, hiszen pl.  $f(n) = n^2$  kielégíti a feltételt, de  $f \notin O(n)$ .

(c) Mivel az  $O(n^2)$  feltétel szerint a felső becslésnek egy megfelelő  $c$ -re és valamely  $n_0$ -nál nagyobb  $n$ -ekre kell igaznak lennie de az  $n_0$ -nál rövidebb bemenetekre semmilyen feltételt nem ad, tehát lehetséges. (Eleg nagy  $c$  választásával is lehet garantálni, hogy egy adott  $|x|$ -re  $10|x|^2 \log |x| - 800 \leq c|x|^2$  teljesüljön.)

3. Egy  $\mathcal{A}$  algoritmról 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?
- (Itt  $|x|$  az  $x$  szó hosszát jelöli.)

*Megoldás:* Jelölje  $f(n)$  az algoritmus maximális lépésszámát az  $n$  hosszú bemeneteken. A feltétel szerint  $f(n) \in O(n \log n)$ , azaz elég nagy  $n$  értékekre  $f(n) \leq cn \log n$  teljesül valamilyen pozitív  $c$ -re.

(a) Lehetséges, pl.  $|x| \leq n_0$  esetén, vagy ha  $c$  olyan, hogy erre az  $x$ -re  $|x|^2 / \log |x| \leq c$ .

(b) Igen, hiszen pl.  $f(n) = n$  is eleget tesz a feltételnek.

4. Jelölje egy algoritmus maximális lépésszámát az  $n$  hosszú bemeneteken  $L(n)$ . Tudjuk, hogy minden  $n > 3$  egész számra  $L(n) \leq L(n-1) + \frac{n}{2}$ , és hogy  $L(3) = 3$ . Következik-e ebből, hogy az algoritmus lépésszáma  $O(n^2)$ ?

1. *Megoldás:* A megadott rekurziós képletet használva  $L(n)$ -re,  $L(n-1)$ -re, stb. kapjuk, hogy  $L(n) \leq L(n-1) + \frac{n}{2} \leq L(n-2) + \frac{n-1}{2} + \frac{n}{2} \leq \dots \leq L(3) + \frac{4}{2} + \frac{5}{2} + \dots + \frac{n}{2} = 3 + \frac{n(n+1)}{4} - \frac{1+2+3}{2} = \frac{1}{4}n^2 + \frac{1}{4}n \in \Theta(n^2)$ , tehát az állítás igaz.

2. *Megoldás:* Bizonyítsuk teljes indukcióval:  $n = 3$ -ra  $L(3) = 3 \leq cn^2$  teljesül, minden  $c \geq 1/3$  esetén.

Ha már tudjuk, hogy egy adott  $c$ -re  $L(n-1) \leq c(n-1)^2$ , akkor  $n > 3$

esetén  $L(n) \leq L(n-1) + \frac{n}{2} \leq c(n-1)^2 + \frac{n}{2} = cn^2 - n(2c-1/2) + c \leq cn^2 - (c-1/2)n \leq cn^2$ , ha  $c \geq 1/2$ . Tehát pl.  $c = 1/2$ ,  $n_0 = 3$  jó választás.

5. 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  $O(n)$ ?

*Megoldás:* Nem következik, hiszen a feltétel semmit nem mond a páratlan hosszú bemenetekről, tehát pl. lehet, hogy minden  $k$  pozitív egészre  $L(2k) = 10$  és  $L(2k+1) = 2^{2k+1}$ .

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

*Megoldás:* (a) A feltételek szerint léteznek olyan  $c_1, c_2 > 0$ ,  $n_0, n_1$  számok, hogy  $|f(n)| \leq c_1|g(n)|$  ha  $n \geq n_0$  és  $|g(n)| \leq c_2|h(n)|$  ha  $n \geq n_1$ . Ezeket összerakva kapjuk, hogy  $|f(n)| \leq c_1|g(n)| \leq c_1c_2|h(n)|$  ha  $n \geq \max(n_1, n_2)$ , tehát az állítás igaz. ( $c = c_1 \cdot c_2$ ,  $n_0 = \max(n_1, n_2)$ ).

(b) Az előzőhöz hasonlóan ellenőrizhető, hogy ez is igaz.