Suport de curs:

Introduction to Algorithms - Cormen

Nota finală: $\frac{2}{3}$ examen + $\frac{1}{3}$ laborator

E-mail: alexandru.popa@fmi.unibuc.ro

alexpopa.neocities.org

Notații asimptotice: $o, O, \Theta, \Omega, \omega$

Când proiectăm algoritmi avem în vedere:

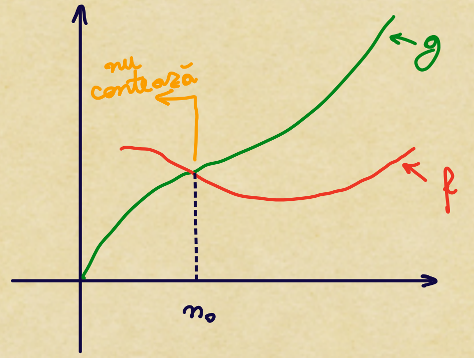
- corectitudinea
- eficiența ⟨ spațiu de memorie / timp de rulare

Timpul de rulare = numărul de operații

Notații asimptotice: o ("o" mic), O ("O" mare), Θ (Theta), Ω (omega), ω (omega mic)

I. O ("O" mare)

$O(g)$ = (informal) mulțimea funcțiilor care cresc mai încet sau la fel de încet ca g



Ex: Un algoritm de complexitate $O(n^2)$ = funcția corespunzătoare nr-ului de pași executați de algoritm crește mai încet ca n^2

Definiție:

Spunem că o funcție $f \in O(g)$ dacă

$\exists\, n_0,\ c > 0$ a.î. $\forall\, n \geq n_0$ avem $f(n) \leq c \cdot g(n)$

Ex: • $2n \in O(n^2)$

$f(n) = 2n$

$g(n) = n^2$

$n_0 = 2$

$c = 1$

$\forall\, n \geq 2 \quad 2n \leq 1 \cdot n^2$

• $200n^2 \in O(n^2)$ Adevărat

$n_0 = 1$

$c = 200$

$\forall\, n \geq 1 \quad 200n^2 \leq c \cdot n^2$

• $n^2 \notin O(n)$

Presupunem prin reducere la absurd că $n^2 \in O(n) \Rightarrow$

$\Rightarrow \exists\, n_0,\ c > 0$ a.î. $n^2 < c \cdot n \quad \forall\, n \geq n_0$

$n < c$ contradicție pentru că c este constantă

• $2^{n+1} \in O(2^n)$?

$c = 2$

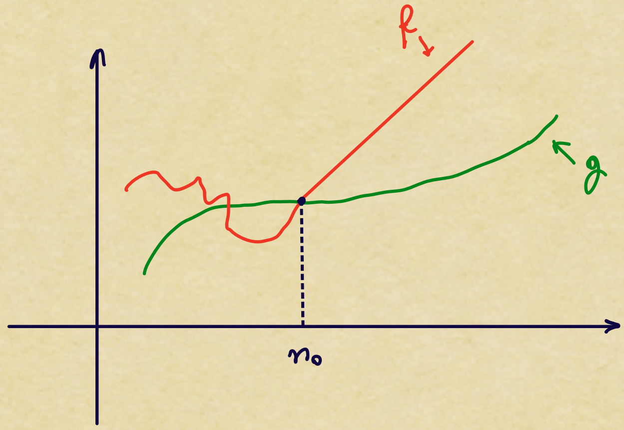
$n_0 = 1$

DA

• $2^{2n} \in O(2^n)$? NU

Temă / Ex la Seminar

II. $\Omega(g)$ = (informal) mulțimea funcțiilor care cresc mai repede sau la fel de repede ca g



Ex: $n^2 \in \Omega(n \log n)$

$\frac{n}{2} \in \Omega(n)$

$2^n \in \Omega(\log n)$

<u>Definiție</u>: $f \in \Omega(g)$ dacă $\exists\, n_0,\ c > 0$ a.î. $\forall\, n \geq n_0$ avem $f(n) \geq c \cdot g(n)$

III. Θ $\quad f \in \Theta(g)$ = (informal) f crește la fel de repede ca g

<u>Definiție</u> $f \in \Theta(g)$ dacă $\exists\, c_1, c_2, n_0 > 0$ a.î. $\forall\, n > n_0$ avem $c_1 g(n) \leq f(n) \leq c_2 \cdot g(n)$

Ex: $200n^2 \notin \Theta(n)$

$200n^2 \in \Theta(n^2)$

$\frac{n^2}{10} \in \Theta(n^2)$

$n \notin \Theta(n \log n)$

<u>Teoremă</u>: $f \in \Theta(g) \iff f \in O(g)$ și $f \in \Omega(g)$

(seminar / exerciții)

IV. $f \in o(g)$ = (informal) f crește **strict** mai încet decât g

Ex: $n \in o(n \log n)$

$n^2 \in o(n^3)$

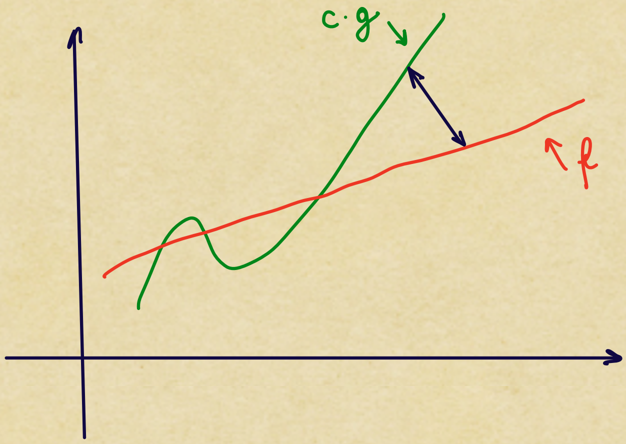
$n \notin o(n)$

$n \notin o(100n)$

$\frac{n}{2} \notin o(n)$

Definiție $f \in o(g)$ dacă $\forall\, c > 0\ \exists\, n_0 > 0$ a.î. $\forall\, n \geq n_0$ avem

$$f(n) < c \cdot g(n)$$



V. $f \in \omega(g)$ = (informal) f crește **strict** mai repede decât g

Ex: $n^3 \in \omega(n^2)$

$2^n \in \omega(n)$

Ex: • $n \in o(n^2)$

Fie $c > 0$ fixat

Vrem să găsim un n_0 a.î. $\forall\, n \geq n_0$ să avem $n < c \cdot n^2$

n_0 depinde de c. De exemplu, putem alege $n_0 = \frac{1}{c}$

$\forall\, n > \frac{1}{c} \quad n < c \cdot n^2$

- $n! \in o(n^n)$ ← exercițiu

- $\log n! \in \Theta(n \log n)$ ← exercițiu