BRZINA RASTA FUNKCIJA VREMENSKE SLOŽENOSTI (1.)

ASIMPTOTSKA NOTACIJA

Notacija

def. (veliko O): → slovo!

Za dvije nenegativne funkcije $f: \mathbb{N} \to \mathbb{R}$ i $g: \mathbb{N} \to \mathbb{R}$ (→ f-je složenosti, → f-je složenosti)

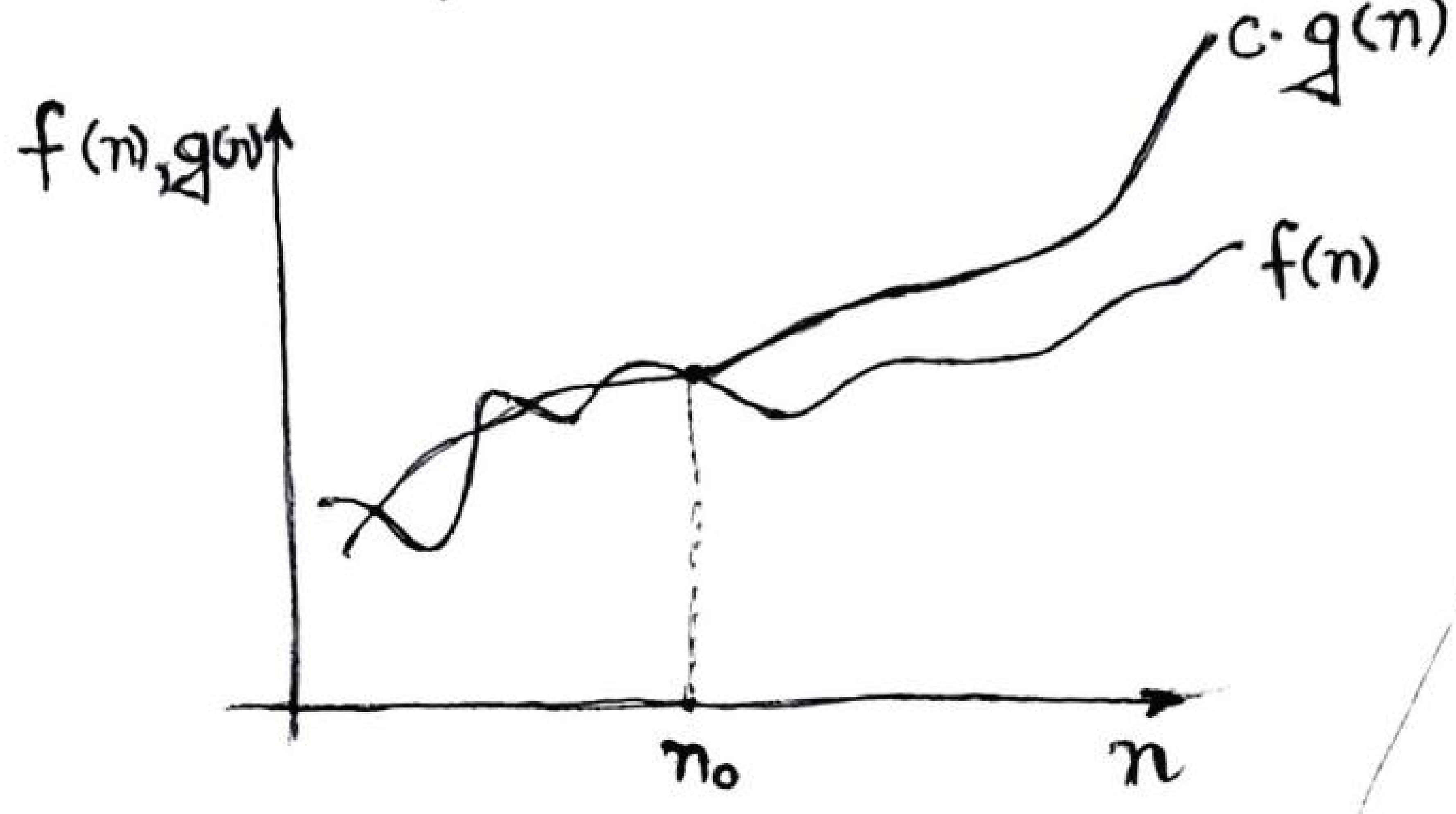
kažemo da je:

$$f(n) = O(g(n))$$

ako postoje pozitivne konstante c i n_0 tako da vrijedi:

$$0 \leq f(n) \leq c \cdot g(n)$$

za svako $n \geq n_0$



$$f(n) = O(g(n))$$

Primjer:

$$f(n) = 3n^2 + 5n + 1$$

uz pp da je uzeto $g(n) = n^2$ =>

$$3n^2 + 5n + 1 \leq c \cdot n^2 \quad / : n^2$$

$$3 + \frac{5}{n} + \frac{1}{n^2} \leq c$$

$$3 + \frac{5}{n_0} + \frac{1}{n_0^2} \leq c$$

$\boxed{\forall n \geq n_0 = 1}$ vrijedi:

$$\frac{1}{n} \leq \frac{1}{n_0} \quad / \cdot 5$$

$$\frac{5}{n} \leq \frac{5}{n_0}$$

$$\frac{1}{n^2} \leq \frac{1}{n_0^2}$$

$$3 + \frac{5}{n} + \frac{1}{n^2} \leq 3 + \frac{5}{n_0} + \frac{1}{n_0^2} = 9$$

za $\forall n \geq n_0 = 1$

$\boxed{\forall n \geq 1}$

$c = 9$

$\boxed{c = 9}$

$$\boxed{c = 9,\ n_0 = 1}$$

npr. ako se uzme $n_0 = 1$ =>

$$c \geq 3 + \frac{5}{1} + \frac{1}{1^2} = 9$$

=> uzmu se const. $n_0 = 1$ i $c = 9$ (mogu i druge:

$n_0 = 2 \Rightarrow c \geq 23/4$

$n_0 = 3 \Rightarrow c \geq 43/9$

...

→ ∞ mnogo njih)

Piše se:

$$f(n) = O(n^2)$$

Može se uzeti i: $g(n) = n^3$, $g(n) = n^4$ ...

n_0	1	2	3	...	$\to\infty$
c	9	$\frac{23}{4}$	$\frac{43}{9}$	...	$\to 3$

Vidi se da ako $n \to \infty \Rightarrow c \to 3$ tj. generalno vrijedi:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$$

Zaključuje se da je notacijom velikog O zapravo definisana gornja granica složenosti! (asimptotska!) → ovim se opisuje vrijeme najgoreg izvršenja!

$f(n)$ je asimptotski ($n \to \infty$) ograničeno sa $c \cdot g(n)$

Nap. • Izražavanje složenosti pomoću notacije O dobija se uvid o ponašanju algoritma za velike vrijednosti n.

Napomena:

Očito je da može postojati beskonačno mnogo fja $g(n)$ za datu funkciju $f(n)$.

Npr. za ovu fju $f(n)$ pored izbora fje za $g(n) = n^2$ moguće je izabrati i fje $g(n) = n^3$; $g(n) = n^4$ itd. ali se uvijek uzima fja sa najmanjim rastom!

Neke osobine notacije O: > ovdje se = simbolički upotrebljava → ne strogo matem. jer ovo ne znači $f_1(n) = f_2(n)$

• Ako vrijedi $f_1(n) = O(g(n))$, $f_2(n) = O(g(n))$ $\Rightarrow$

tada je: $f_1(n) + f_2(n) = O(g(n))$

• Ako vrijedi $f(n) = O(g(n))$ i $g(n) = O(h(n)) \Rightarrow$

tada je: $f(n) = O(h(n))$

• Vrijedi: $an^k = O(n^k)$

• Vrijedi: $\log_a n = O(\log_b n)$, za bilo koje $a, b > 0$ i $b \neq 1$

• Vrijedi: $f(n) = O(f(n))$

Scanned by TapScanner

Notacija → omega!

def. (veliko Ω)

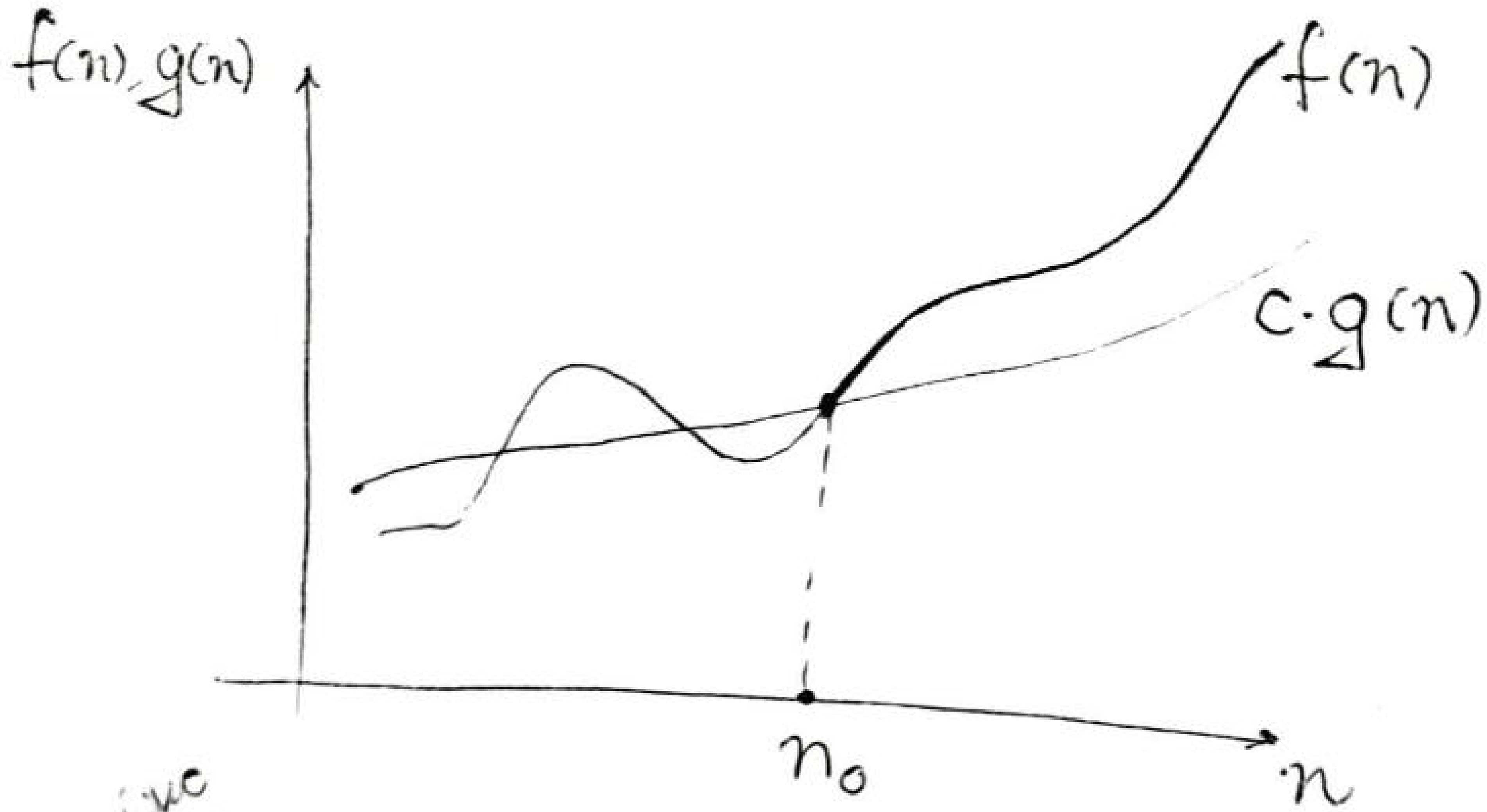
Za dvije nenegativne funkcije $f: \mathbb{N} \to \mathbb{R}$ i $g: \mathbb{N} \to \mathbb{R}$ kažemo da je:

$$f(n) = \Omega(g(n))$$

ako postoje **pozitivne** konstante c i n_0 tako da vrijedi:

$$f(n) \geq c \cdot g(n) \geq 0$$

za svako $n \geq n_0$.



Notacija veliko Ω definiše donju granicu složenosti.

Neke osobine notacije Ω:

- Ako vrijedi $f(n) = \Omega(g(n))$ i $g(n) = \Omega(h(n))$
  tada je: $f(n) = \Omega(h(n))$
- Vrijedi: $f(n) = \Omega(f(n))$

Primjer: $f(n) = 3n^2 + 5n + 1$

pp. opet $g(n) = n^2$

$$3n^2 + 5n + 1 \geq c \cdot n^2 \quad /: n^2$$

$$3 + \frac{5}{n} + \frac{1}{n^2} \geq c$$

$$3 + \underbrace{\frac{5}{n}}_{>0} + \underbrace{\frac{1}{n^2}}_{>0} \geq 3 = c$$

$\forall n \geq n_0 = 1$

$$\boxed{c = 3,\ n_0 = 1}$$

Uzmimo

$\Rightarrow$ $f(n) = \Omega(n^2)$

Isto mogli smo uzeti $g(n) = n$; $g(n) = \sqrt{n}$; ...

notacije — theta

def (veliko Θ)

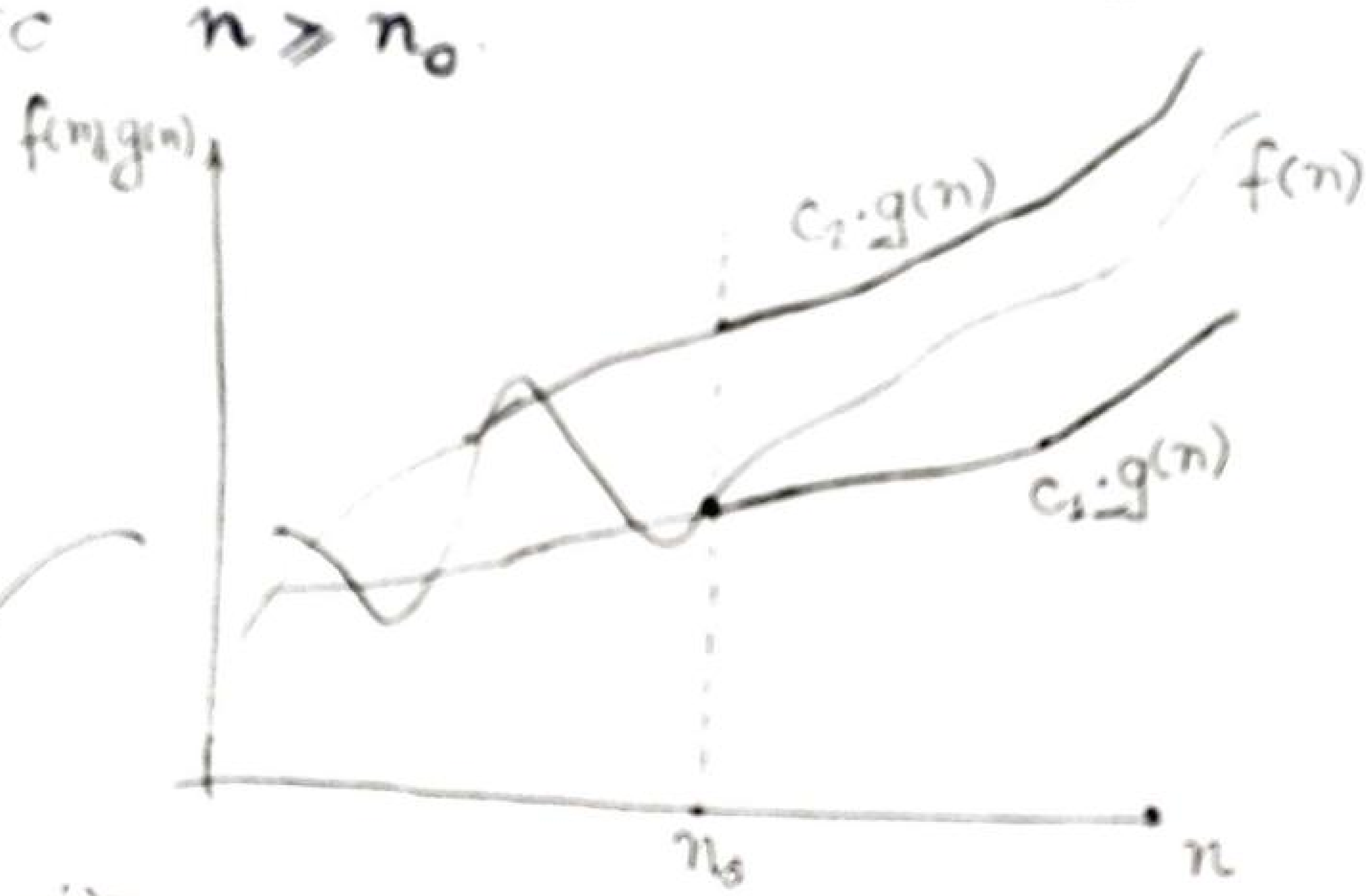
Za dvije nenegativne funkcije $f: N \to R$ i $g: N \to R$ kažemo da je:

$$f(n) = \Theta(g(n))$$

ako postoje pozitivne konstante $c_1 c_2$ i n_0 takve da je:

$$0 \le c_1 \cdot g(n) \le f(n) \le c_2 \cdot g(n)$$

za svako $n \ge n_0$



Notacijom Θ (veliko) se definišu asimptotske granice (složenosti) performansi nekog algoritma i sa gornje i sa donje strane!

Sa ove slike očito je da vrijedi:

$$f(n) = \Theta(g(n)) \quad \text{ako je} \quad f(n) = O(g(n)) \quad \text{i} \quad f(n) = \Omega(g(n))$$

Nap.

Za primjer fje: $f(n) = 3n^2 + 5n + 1$ je veliko O notacija:

$$f(n) = O(g(n))$$

gdje $g(n)$ mogu biti fje n^2, n^3, n^4 itd.

Međutim, jedina od ovih fja $g(n)$ kojom se može izraziti složenost u veliko Ω notaciji: ~~je~~

$$f(n) = \Omega(g(n))$$

je fja $g(n) = n^2$

To znači da u notaciji veliko Θ vrijedi:

$$f(n) = \Theta(n^2)$$

notacija

· def. (malo o)

Za dvije nenegativne funkcije $f: \mathbb{N} \to \mathbb{R}$ i $g: \mathbb{N} \to \mathbb{R}$ kažemo da je:

$$f(n) = o(g(n))$$

ako za svaku pozitivnu konstantu c postoji konstanta n_0 tako da je:

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

za svako $n \geq n_0$.

malo o

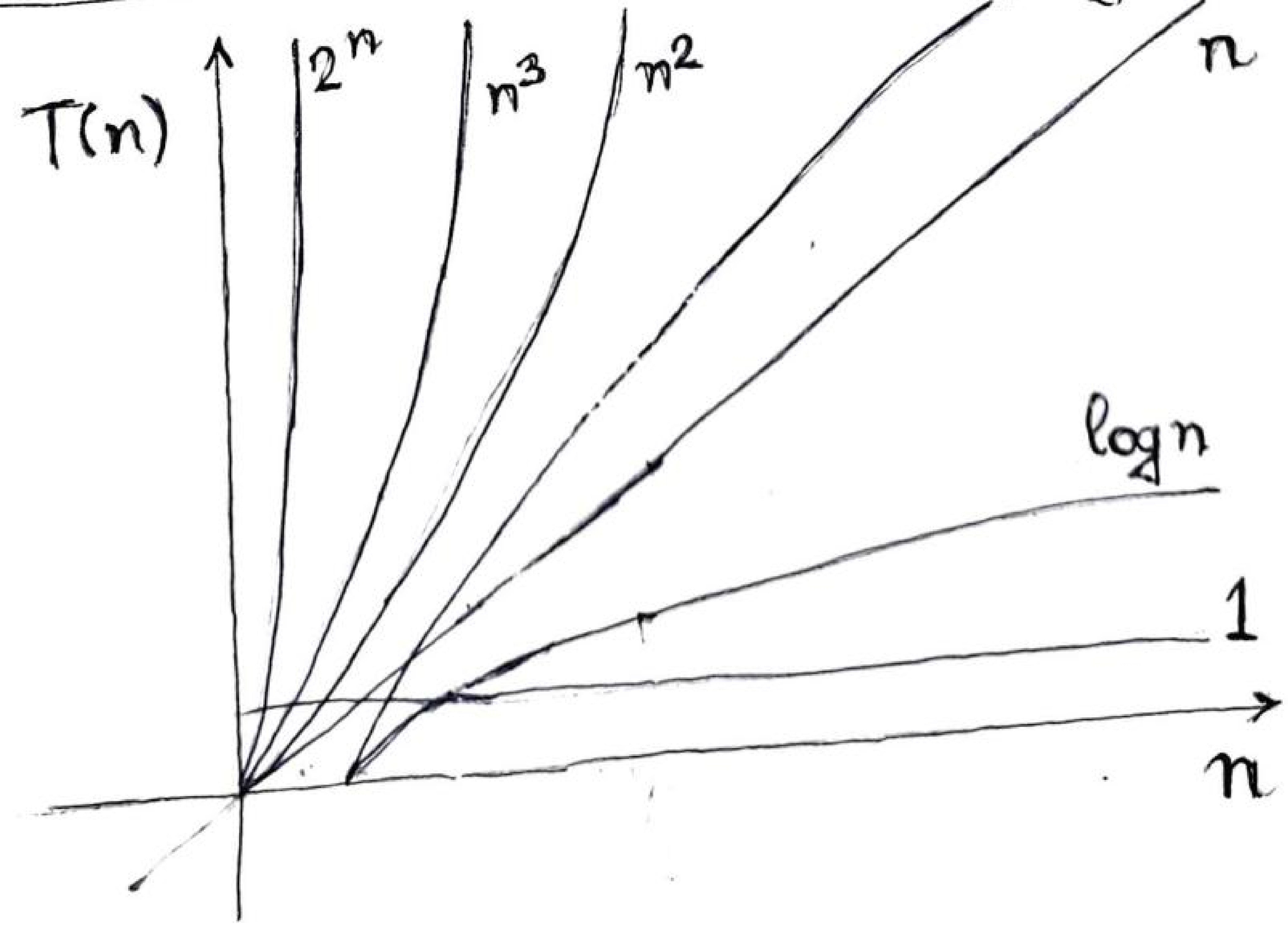
Primjer:

Pokazati da vrijedi: $2n = o(n^2)$ i $2n^2 \neq o(n^2)$

Asimptotska notacija	Relativna brzina rasta
$f(n) = O(g(n))$	rast $f(n) \leq$ rast $g(n)$
$f(n) = \Omega(g(n))$	rast $f(n) \geq$ rast $g(n)$
$f(n) = \Theta(g(n))$	rast $f(n) \approx$ rast $g(n)$
$f(n) = o(g(n))$	rast $f(n) \ll$ rast $g(n)$

Tipične asimptotske funkcije vremena izvršavanja:

Funkcija	ime
1	konstantna funkcija
n	linearna funkcija
n^2	kvadratna funkcija
n^3	kubna funkcija
$\log n$	logaritamska funkcija
2^n	eksponencijalna funkcija
$n \cdot \log n$	$n \cdot \log n$



Primjer: Pokazati da vrijedi:

$$\frac{1}{2}n^2 - 3n = \Theta(n^2)$$

Rješenje: Potrebno je odrediti pozitivne konstante c_1, c_2 i n_0 takve da vrijedi za sve $n \geq n_0$:

$$c_1 n^2 \leq \frac{1}{2}n^2 - 3n \leq c_2 n^2 \quad / : n^2$$

$$c_1 \leq \frac{1}{2} - \frac{3}{n} \leq c_2$$

I: $\frac{1}{2} - \frac{3}{n} \leq c_2$

$$\underbrace{\frac{1}{2} - \frac{3}{n} \leq \frac{1}{2}}_{\text{za sve } n \geq n_0 = 1} = c_2$$

Dakle, desna nejednakost vrijedi za $c_2 = \frac{1}{2}$; $n_0 = 1$ $(n \geq n_0 = 1)$

II: $c_1 \leq \frac{1}{2} - \frac{3}{n}$

c_1 mora biti pozitivna const. $\Rightarrow$

$$\frac{1}{2} - \frac{3}{n} > 0 \Rightarrow \frac{1}{2} > \frac{3}{n} \quad / \cdot 2n \Rightarrow n > 6 \Rightarrow$$

$$\Rightarrow n \geq n_0 = 7$$

$$n \geq n_0 \quad /(\,)^{-1}$$

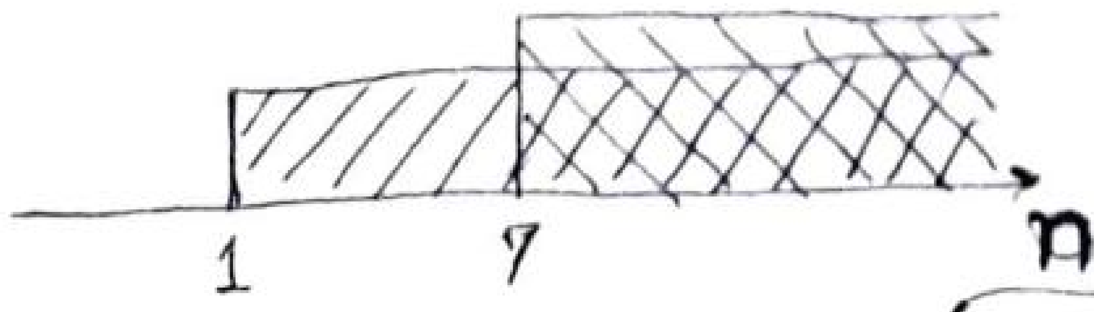
$$\frac{1}{n} \leq \frac{1}{n_0} \quad / \cdot (-3)$$

$$-\frac{3}{n} \geq -\frac{3}{n_0} \quad / + \frac{1}{2}$$

$$\frac{1}{2} - \frac{3}{n} \geq \frac{1}{2} - \frac{3}{n_0} = c_1$$

$$n_0 = 7 \Rightarrow c_1 = \frac{1}{2} - \frac{3}{7} = \frac{7-6}{14} = \frac{1}{14} \Rightarrow c_1 = \frac{1}{14}$$

Dakle, lijeva strana nejednakosti vrijedi za $c_1 = \frac{1}{14}$; $n_0 = 7$ $(n \geq n_0 = 7)$



$$\Rightarrow \boxed{n_0 = 7;\ c_1 = \frac{1}{14},\ c_2 = \frac{1}{2}}$$

8.

.. Primjer: Pokazati da je tačno:

$$6n^3 \neq \Theta(n^2)$$

Pp. suprotno: postoje konstante c_1, c_2 i n_0:

$$\underbrace{c_1 n^2 \leq 6n^3}_{\text{ovo je OK!}} \leq c_2 n^2 \ ?$$

$$6n^3 \leq c_2 n^2 \ / :6n^2 \quad \text{za} \quad n \geq n_0$$

$$n \leq \frac{c_2}{6} \rightarrow$$ što je $\perp$ jer je c_2 const. dok je n proizvoljno velik prirodni broj $(n \rightarrow \infty)$.

DZ. Potrebno je formalnom definicijom notacije dokazati sljedeće asimptotske relacije: (9.)

ⓐ $n^3 + n^2 - 1 = \Theta(n^3)$

ⓑ $2n^2 + n - 1 = O(n^2)$

ⓒ $n \cdot \log n = O(n^2)$

ⓓ $\log n = O(\sqrt{n})$

ⓔ $17 n \log n - 23n - 10 = \Theta(n \log n)$

ⓕ $n^{\log n} = O(2^n)$

ⓖ $\sin n = O(1)$

Tabela 2.5

Klasa / veličina ulaza n	$O(1)$	$O(\log n)$	$O(n \log n)$	$O(n^2)$	$O(n^3)$	$O(2^n)$
10	1	3.32	33.2	10^2	10^3	1024
	0.1 μsec	0.3 μsec	3.3 μsec	10 μsec	100 μsec	1 msec
10^2	1	6.64	664	10^4	10^6	10^{30}
	0.1 μsec	0.7 μsec	66.4 μsec	1 msec	0.1 sec	$3.17 \cdot 10^{15}$ god.
10^3	1	9.97	$9.97 \cdot 10^3$	10^6	10^9	10^{301}
	0.1 μsec	1 μsec	1 msec	0.1 sec	1.67 min	$3.17 \cdot 10^{286}$ god.
10^4	1	13.3	$133 \cdot 10^3$	10^8	10^{12}	10^{3010}
	0.1 μsec	1.3 μsec	13.3 msec	0.17 min	1.16 dana	
10^5	1	16.6	$1.66 \cdot 10^6$	10^{10}	10^{15}	10^{30103}
	0.1 μsec	1.7 μsec	0.16 sec	16.7 min	3.17 god.	
10^6	1	19.93	$19.93 \cdot 10^6$	10^{12}	10^{18}	10^{301030}
	0.1 μsec	2 μsec	2 sec	1.16 dana	3170.9 god.	

Iz tabele 2.5 možemo vidjeti da npr. algoritmi složenosti $O(n^3)$ za veličinu ulaza 10^5 praktično imaju samo teoretski značaj, jer je potrebno više od 3 godine da bi se algoritam te složenosti izveo. Ili, na primjer, za veličinu ulaza 10^6, potrebno je više od 3170 godina. Još kompleksnija situacija je sa eksponencijalnim algoritmima, koji imaju složenost npr. $O(2^n)$. Iz tabele 2.5 možemo vidjeti da je već za veličinu ulaza 10^3 potrebno izvršiti 10^{301} operacija, dok bi za izvođenje tolikog broja operacija bilo potrebno $3.17 \cdot 10^{286}$ godina. To je nezamislivo mnogo vremena. Dakako, još kompleksnija situacija je za veličine ulaza 10^5 ili 10^6. Da bi dobili dojam o veličini nekih vremenskih intervala prikazanih u tabeli 2.5, spomenimo npr. da je starost planete Zemlje oko $4{,}5 \cdot 10^9$ godina. Ili npr. procjenjuje se da je ukupan broj atoma u svemiru između 10^{78} i 10^{81}.