Algoritmi in podatkovne strukture 1

Visokošolski strokovni študij Računalništvo in informatika

Asimptotična zahtevnost

Jurij Mihelič, UniLj, FRI

Določanje zahtevnosti

- **Natančna zahtevnost** algoritma
  - težavno določanje
    - upoštevanje **podrobnosti** algoritma
    - upoštevanje dejanskih **vhodnih podatkov**
    - poznavanje **stroška** posameznih ukazov
    - natančna zahtevnost je pogosto »*grda*« funkcija

      $T(n) = 4/3n^3 + 2\sqrt{n} \lg n - 16 \lg n + 42$
  - omejena uporabnost
    - **številčnost in raznolikost** modelov računanja / računalniških arhitektur / programskih jezikov
    - v **praksi** na zahtevnost (čas izvajanja) vpliva tudi operacijski sistem in drugi (sistemski) programi

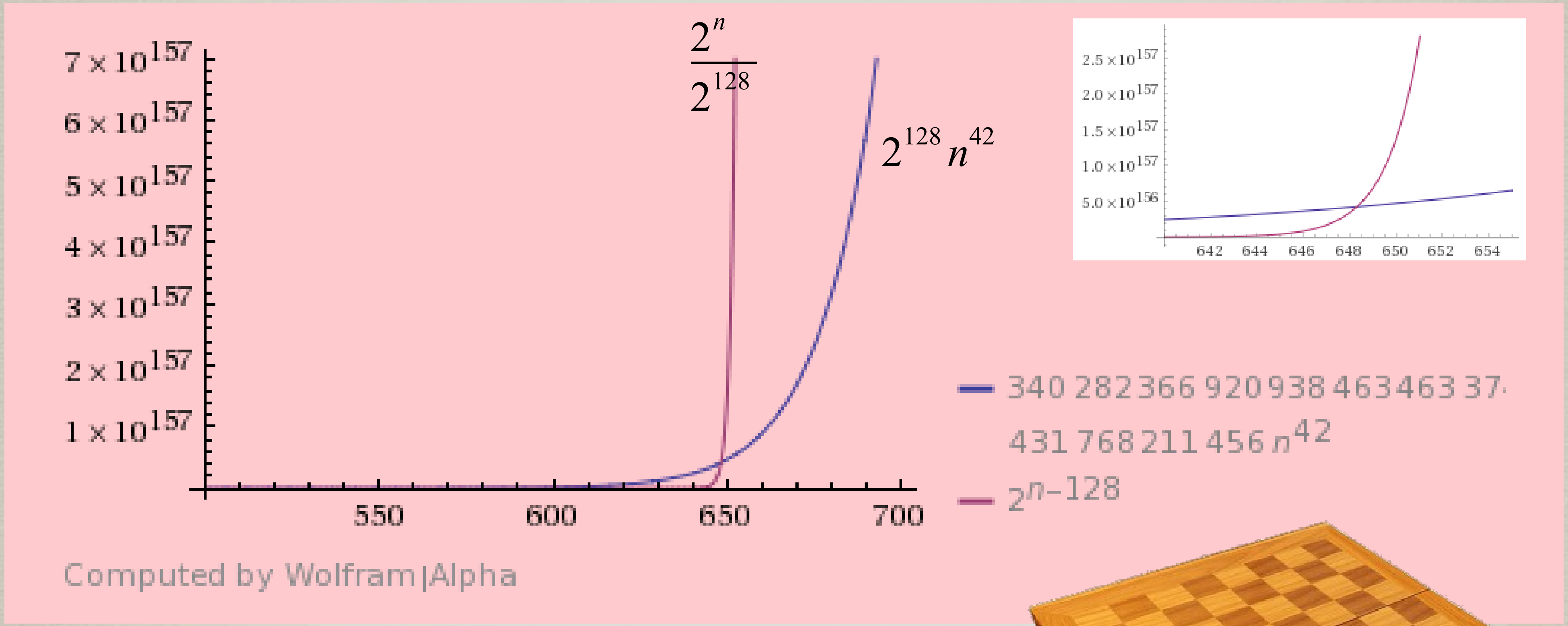
Zahtevnosti pogosto ne znamo/moremo/želimo natančno določiti.

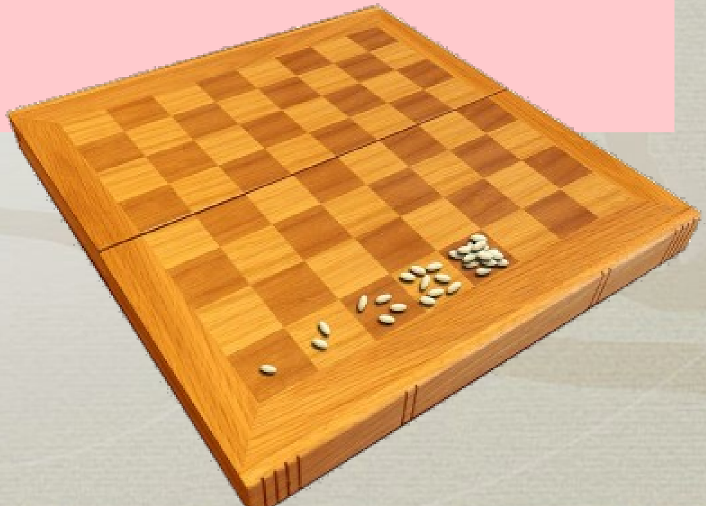
Določanje zahtevnosti

- Asimptotična **ocena zahtevnosti**
  - opazujemo naraščanje zahtevnosti glede na naraščanje velikosti naloge
    - asimptotični pogled, glede na velike naloge
    - velikost naloge $n \to \infty$
  - ocena zahtevnosti je »*lepa*« funkcija
    - $T(n)$ je reda n^3
    - $T(n)$ narašča hitreje kot n^2
    - $T(n)$ ne narašča hitreje kot n^4

Asimptotična zahtevnost

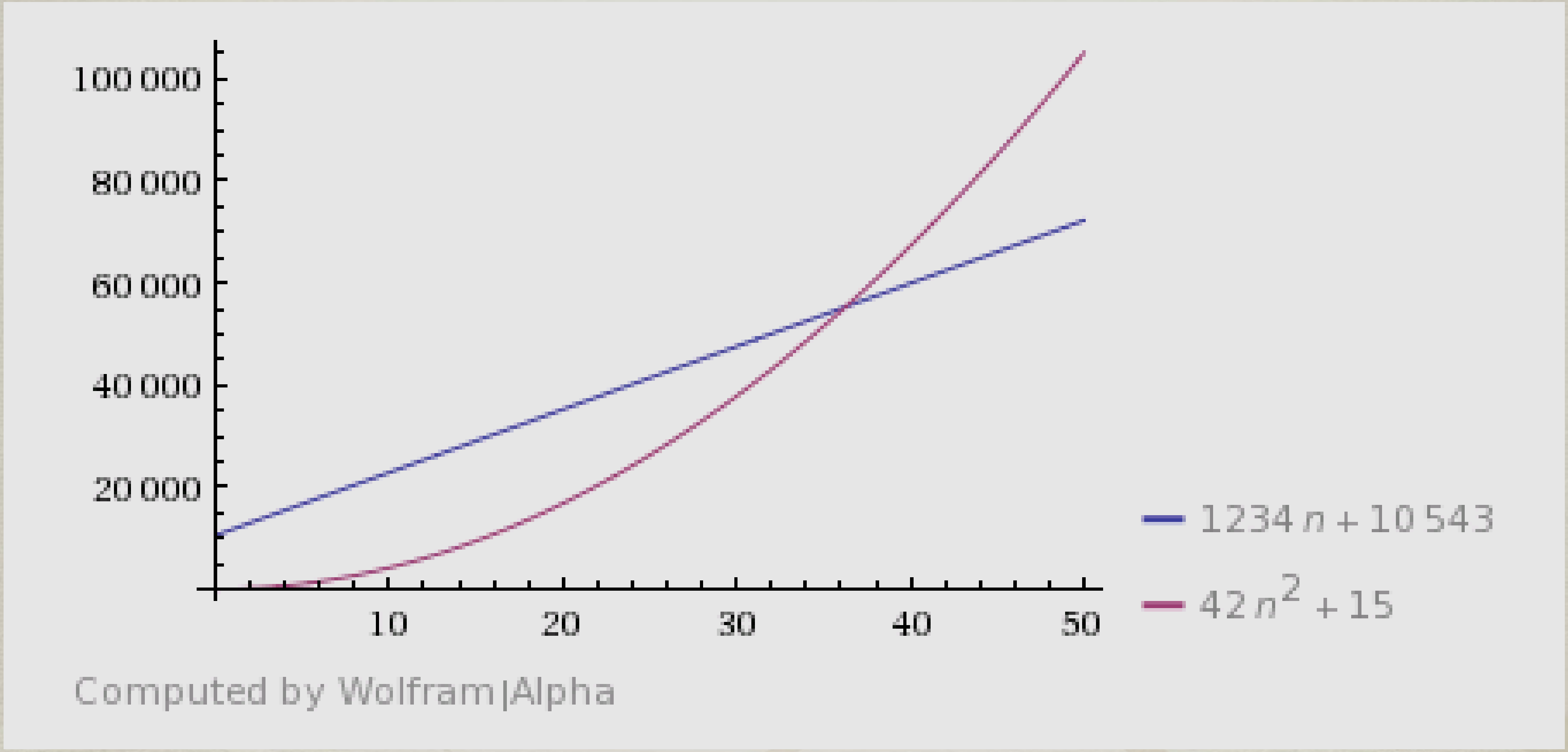
- Zakaj $n \to \infty$?





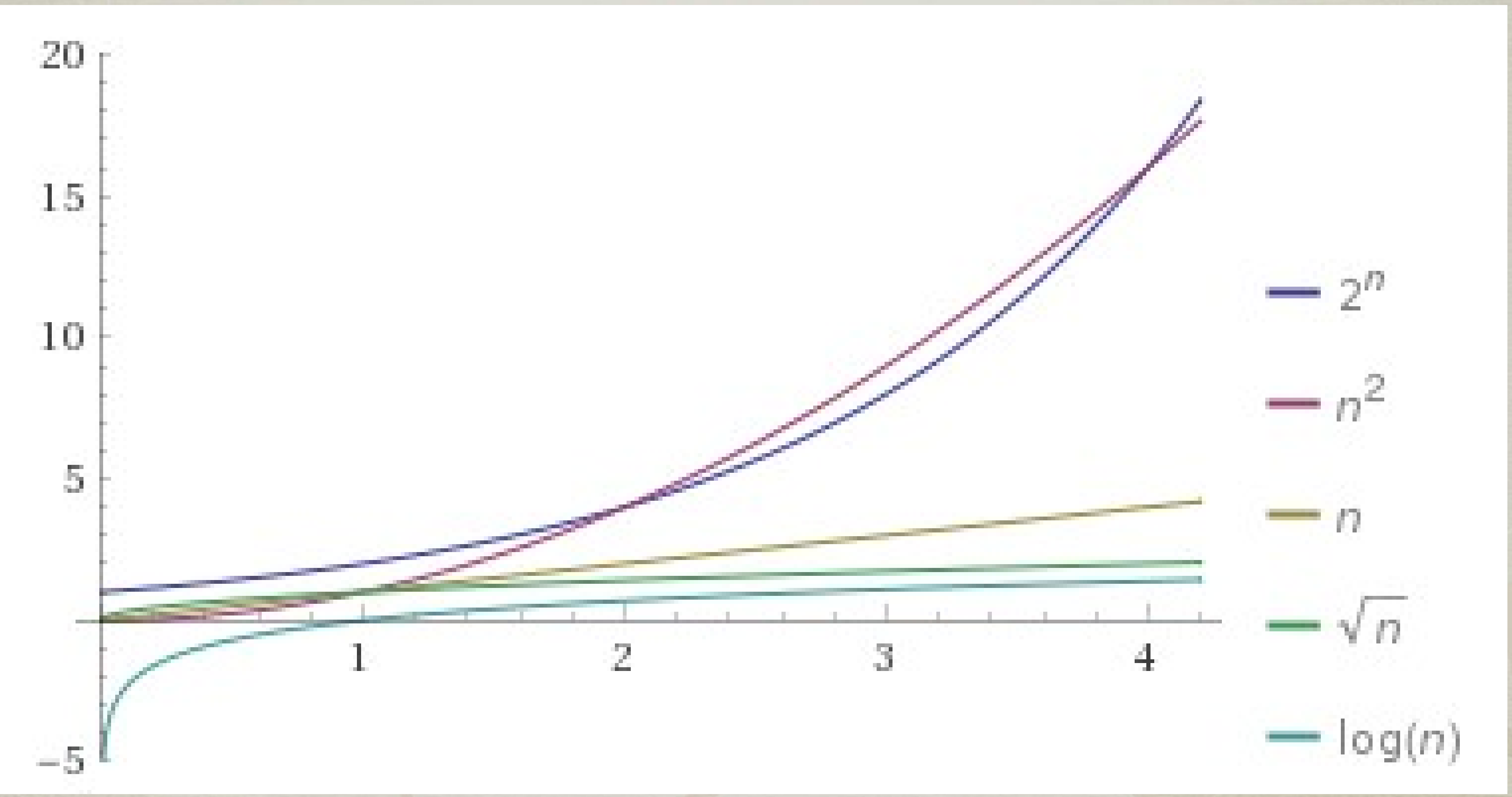
Asimptotična zahtevnost

- Zakaj $n \to \infty$?



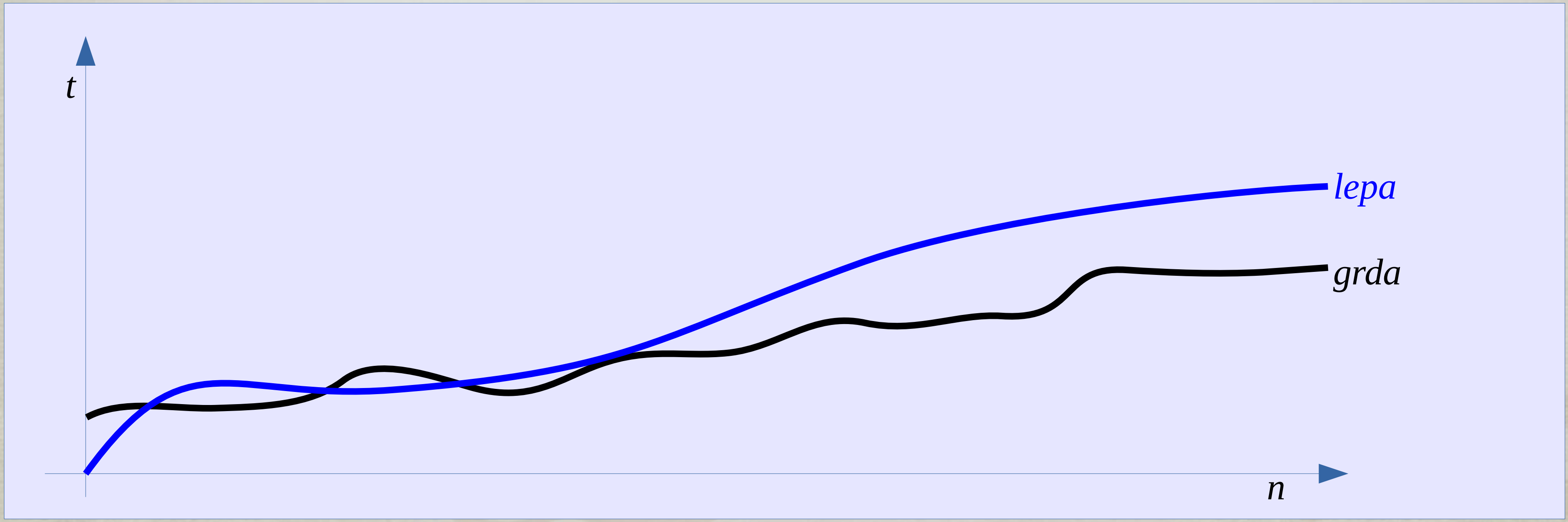
Asimptotična zahtevnost

- Zakaj $n \to \infty$?



Asimptotična notacija

- Funkcije lahko ocenjujemo
  - od zgoraj
  - od spodaj
  - od zgoraj in spodaj

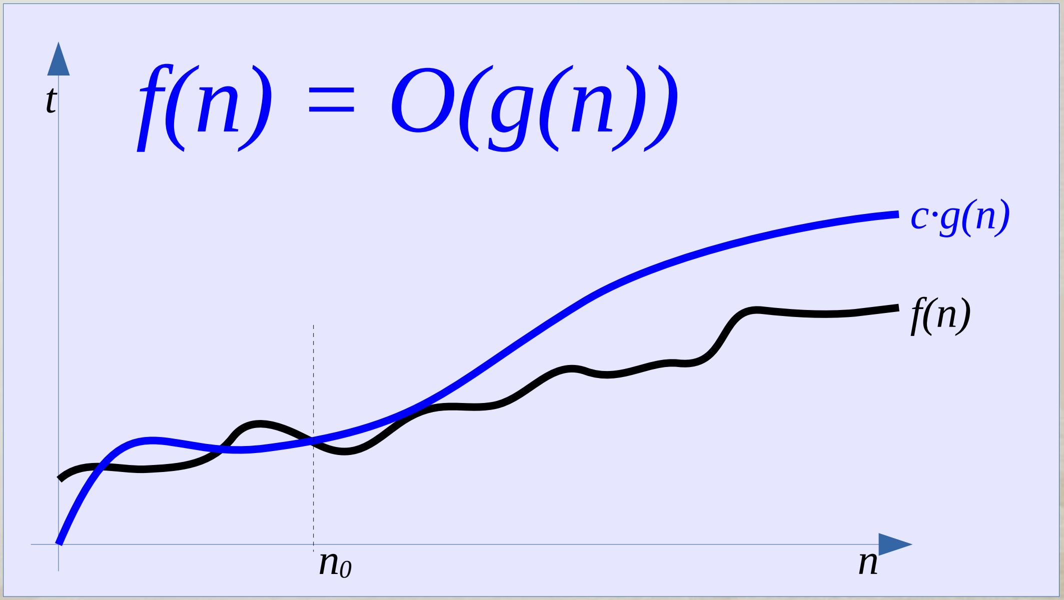


Asimptotična notacija

- O -notacija – **zgornja** asimptotična meja
  - f je od zgoraj omejena z g
  - f narašča kvečjemu tako hitro kot g
  - f ne raste hitreje kot g
  - f je kvečjemu reda g

$$
\begin{aligned}
5n^2 + 12 &\neq O(\log n) \\
&\neq O(n) \\
&\neq O(n^{1.9}) \\
&\mathbf{= O(n^2)} \\
&= O(n^3) \\
&= O(1.001^n) \\
&= O(2^n)
\end{aligned}
$$

*Katera izmed zgornjih mej je **tesna**?*



Omega

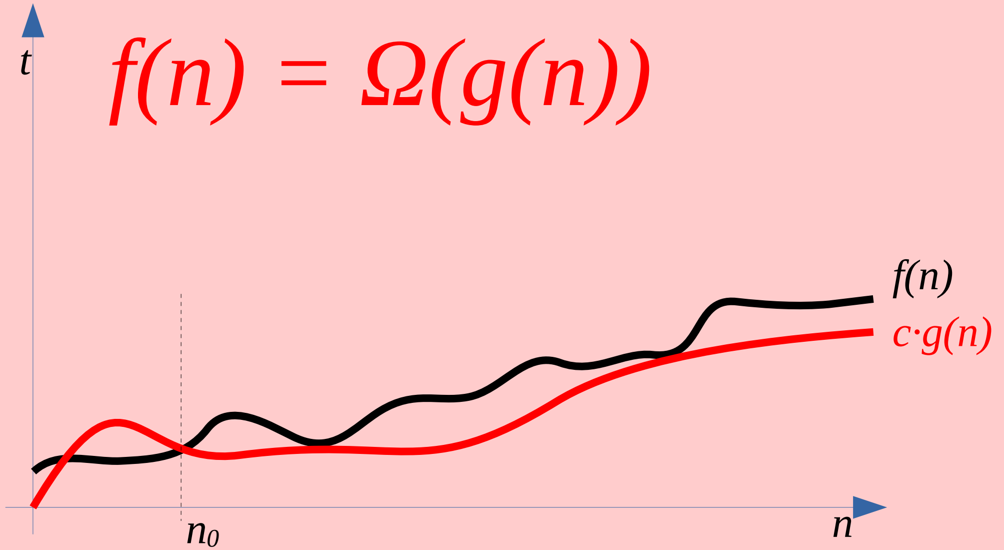
Asimptotična notacija

- Ω -notacija – **spodnja** asimptotična meja
  - f je od spodaj omejena z g
  - f narašča vsaj tako hitro kot g
  - f ne raste počasneje kot g
  - f je vsaj reda g

$$
\begin{aligned}
5n^2 + 12 &= \Omega(\log n) \\
&= \Omega(n) \\
&= \Omega(n^{1.9}) \\
&\mathbf{= \Omega(n^2)} \\
&\neq \Omega(n^3) \\
&\neq \Omega(1.001^n) \\
&\neq \Omega(2^n)
\end{aligned}
$$

*Katera izmed spodnjih mej je **tesna**?*

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

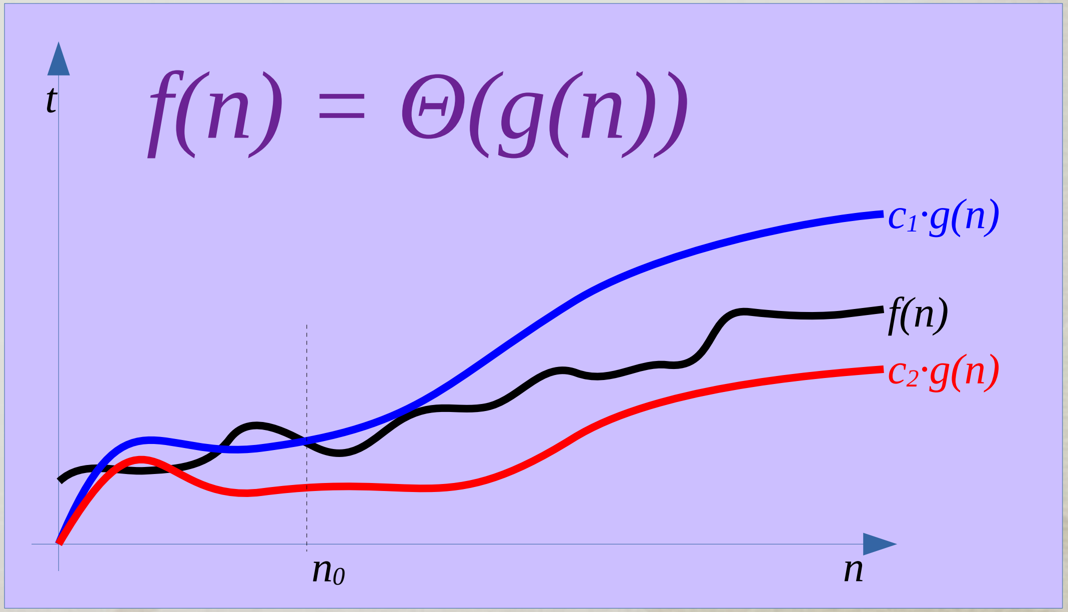


Téta

Asimptotična notacija

- *Θ* -notacija – **tesna** asimptotična meja
  - f je od zgoraj in od spodaj omejena z g
  - f je reda g

$$5n^2 + 12 \neq \Theta(\log n)$$
$$\neq \Theta(n)$$
$$\neq \Theta(n^{1.9})$$
$$= \Theta(n^2)$$
$$\neq \Theta(n^3)$$
$$\neq \Theta(1.001^n)$$
$$\neq \Theta(2^n)$$



Asimptotična notacija

Množice

Donald E. Knuth

- Formalne definicije
  - O, Ω, Θ

$$O(g(n)) = \{ f(n) \mid \exists c, n_0 > 0 \; \forall n \geq n_0 : 0 \leq f(n) \leq c g(n) \}$$

$$\Theta(g(n)) = \{ f(n) \mid \exists c_1, c_2, n_0 > 0 \; \forall n \geq n_0 : 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \}$$

$$\Omega(g(n)) = \{ f(n) \mid \exists c, n_0 > 0 \; \forall n \geq n_0 : 0 \leq c g(n) \leq f(n) \}$$

Vse funkcije so nenegativne.

Dodatna asimptotična notacija

- Dodatni notaciji
  - *o* notacija: f narašča počasneje od g
  - *ω* notacija: f narašča hitreje od g

<	$o(g(n))=\{f(n) \mid \forall c>0\, \exists n_0\, \forall n\geq n_0 : 0\leq f(n)<cg(n)\}$
≤	$O(g(n))=\{f(n) \mid \exists c, n_0>0\, \forall n\geq n_0 : 0\leq f(n)\leq cg(n)\}$
=	$\Theta(g(n))=\{f(n) \mid \exists c_1, c_2, n_0>0\, \forall n\geq n_0 : 0\leq c_1 g(n)\leq f(n)\leq c_2 g(n)\}$
≥	$\Omega(g(n))=\{f(n) \mid \exists c, n_0>0\, \forall n\geq n_0 : 0\leq cg(n)\leq f(n)\}$
>	$\omega(g(n))=\{f(n) \mid \forall c>0\, \exists n_0\, \forall n\geq n_0 : 0\leq cg(n)<f(n)\}$

Dodatna asimptotična notacija

- Notacija s ~ (tildo)
  - funkciji naraščata enako hitro

$$f(n) \sim g(n) \equiv \lim_{n \to \infty} \frac{f(n)}{g(n)} = 1$$

Robert Sedgewick

- Intuitivno
  - ujemanje v redu velikosti
  - ujemanje v konstanti
  - kot Θ-notacija s konstanto
  - npr.: $5n^3 + 2n^2 + n + \lg n \sim 5n^3$

Definicije z limito

- Z limitami

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

namig	notacija	limita
$<$	o	$L = 0$
$\leq$	O	$0 \leq L < \infty$
$=$	Θ	$0 < L < \infty$
$\geq$	Ω	$0 < L \leq \infty$
$>$	ω	$L = \infty$
$\sim$	$\sim$	$L = 1$

Pri računanju limit pride prav l'Hopitalovo pravilo

0	1	L	∞
o, O	$\sim$	Θ, O, Ω	ω, Ω

Uporaba asimptotične notacije

- Zahtevnost algoritma
  - natančne zahtevnost $T(n)$ pri poljubni nalogi velikosti n pogosto ne znamo določiti
  - zato jo asimptotično ocenimo

$$T_{BC}(n) \leq T(n) \leq T_{WC}(n)$$

$$T_{BC}(n) = \Omega(f(n))$$

$$T_{WC}(n) = O(g(n))$$

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

$$T(n) = O(g(n))$$

Uporaba asimptotične notacije

- Ustaljena (zlo-)raba
  - namesto ∈ uporabljamo =

$$f(n)=O(g(n)) \equiv f(n)\in O(g(n))$$

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

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

  - Leva za vse / desna za enega

$$2n^2+3n+1=2n^2+\Theta(n)=\Theta(n^2)$$

Uporaba asimptotične notacije

- Zahtevnost algoritma
  - $T(n) = O(f(n))$ ali $S(n) = O(f(n))$

Funkcija $f(n)$	Zahtevnost algoritma
1	konstantna
$\log n$	logaritmična
$\log^d n$, kjer $d=O(1)$	polilogaritmična
n	linearna
$n \log n$	linearitmična, n-log-n
n^2	kvadratna
n^3	kubična
$n^{O(1)}$ oz. n^d, kjer $d=O(1)$	polinomska
2^n	eksponentna
$n!$	faktoriela

Moorov zakon

- Čas pri podvojeni nalogi
- Velikost naloge pri podvojeni hitrosti

$T(n)$	$T(2n)$	velikost za $2v$
lg n	$T(n) + 1$	n^2
n	$2\,T(n)$	$2n$
$n \lg n$	$2\,T(n) + 2n$	$\sim 2n$
n^2	$4\,T(n)$	$\sqrt{2}n = 1{,}41n$
n^3	$8\,T(n)$	$\sqrt[3]{2}n = 1{,}26n$
2^n	$2^n\,T(n)$	$n + 1$

Kaj pa pri 10×?

Osnovne lastnosti

- Refleksivnost

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

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

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

$$f(n) \sim f(n)$$

$$f(n) \neq o(f(n))$$

$$f(n) \neq \omega(f(n))$$

Osnovne lastnosti

- Simetrija

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

$$f(n)=\Theta(g(n)) \Leftrightarrow f(n)=\Omega(g(n)) \wedge f(n)=O(g(n))$$

$$f(n)\sim g(n) \Leftrightarrow g(n)\sim f(n)$$

- Transponirana simetrija

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

$$f(n)=o(g(n)) \Leftrightarrow g(n)=\omega(f(n))$$

Osnovne lastnosti

- Tranzitivnost
  - velja za vse O, Ω, Θ, o, ω in ~

$$f(n)=O(g(n)) \wedge g(n)=O(h(n)) \Rightarrow f(n)=O(h(n))$$

$$f(n)=\Omega(g(n)) \wedge g(n)=\Omega(h(n)) \Rightarrow f(n)=\Omega(h(n))$$

$$f(n)=\Theta(g(n)) \wedge g(n)=\Theta(h(n)) \Rightarrow f(n)=\Theta(h(n))$$

$$f(n)=o(g(n)) \wedge g(n)=o(h(n)) \Rightarrow f(n)=o(h(n))$$

$$f(n)=\omega(g(n)) \wedge g(n)=\omega(h(n)) \Rightarrow f(n)=\omega(h(n))$$

$$f(n)\sim g(n) \wedge g(n)\sim h(n) \Rightarrow f(n)\sim h(n)$$

Osnovne lastnosti

- Eliminacija konstante
  - naj bo $c > 0$ konstanta, potem

$$O(c \cdot f(n)) = O(f(n))$$

$$\Omega(c \cdot f(n)) = \Omega(f(n))$$

$$\Theta(c \cdot f(n)) = \Theta(f(n))$$

Osnovne lastnosti

- Produkt
  - in gnezdenje zank

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

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

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

Osnovne lastnosti

- Vsota
  - in zaporedni programski bloki

$$O(f(n))+O(g(n))=O(max(f(n),g(n)))$$

$$O(f(n))+O(g(n))=O(g(n)), če\ f(n)=O(g(n))$$

$$\Omega(f(n))+\Omega(g(n))=\Omega(min(f(n),g(n)))$$

$$\Omega(f(n))+\Omega(g(n))=\Omega(g(n)), če\ f(n)=\Omega(g(n))$$

Osnovne lastnosti

- Ostali odnosi

$$f(n) = o(g(n)) \Rightarrow f(n) = O(g(n))$$

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

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