O-notace

Bc. Katarína Olejková

KATEDRA INFORMATIKY

UNIVERZITA PALACKÉHO V OLOMOUCI

O-notace

- Presná analýza časovej zložitosti T(n) (doteraz):
  - Počítali sme počet výpočetných krokov, ktoré algoritmus prevedie
  - Nevýhoda – konštantny závisia na konkrétnom pseudokóde/implementácii

- Nepresná analýza časovej zložitosti T(n):
  - Zaujíma nás ako časová zložitosť rastie keď veľkosť vstupu $n \to \infty$
  - Bude nás zaujímať ako rastie najdôležitejší člen – budeme potlačovať zbytočné konštanty
  - Postačujúca aj keď nie je presná, vyzdvihuje to najpodstatnejšie
  - Na túto analýzu budeme používať matematický nástroj **O-notace**

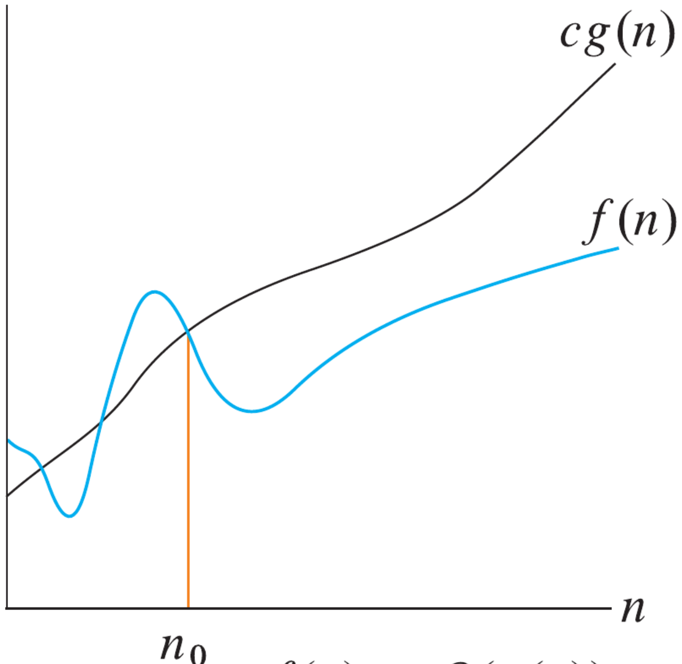
O-notace

- Budeme mať dve funkcie $f(n), g(n)$
  - Nebude nás zaujímať ktorá funkcia je menšia/väčšia pre malé hodnoty n (n – veľkosť vstupu)
  - Nebudú náš zaujímať konštantné faktory rastu funkcie (môžeme funkciu g násobiť)

$O(g)$ ... asymptotická horní mez (odhad)

Definice Pro funkci $g(n)$ je

$$O(g(n)) \quad = \quad \{f(n) \mid \text{ existuje } c > 0 \text{ a } n_0 \in \mathbb{N} \text{ tak,}$$
$$\text{že pro každé } n \geq n_0 \text{ je } 0 \leq f(n) \leq cg(n)\}$$



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

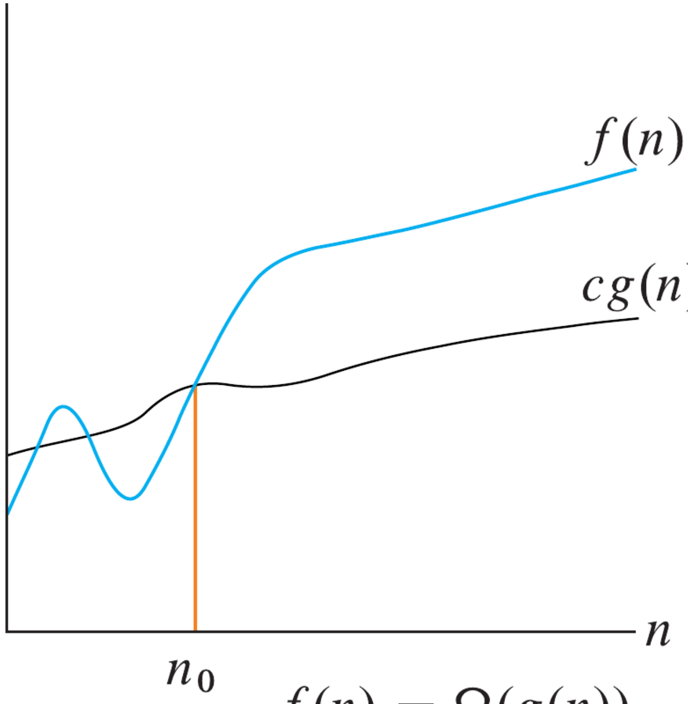
$f \; = O(g)$

f nerastie rýchlejšie ako g

$\Omega(g)$ ... asymptotická dolní mez (odhad)

Definice Pro funkci $g(n)$ je

$$\begin{aligned}\Omega(g(n)) &= \{f(n) \mid \text{ existuje } c > 0 \text{ a } n_0 \in \mathbb{N} \text{ tak,} \\ &\qquad \text{že pro každé } n \geq n_0 \text{ je } 0 \leq cg(n) \leq f(n)\}\end{aligned}$$



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

$f = \Omega(g)$

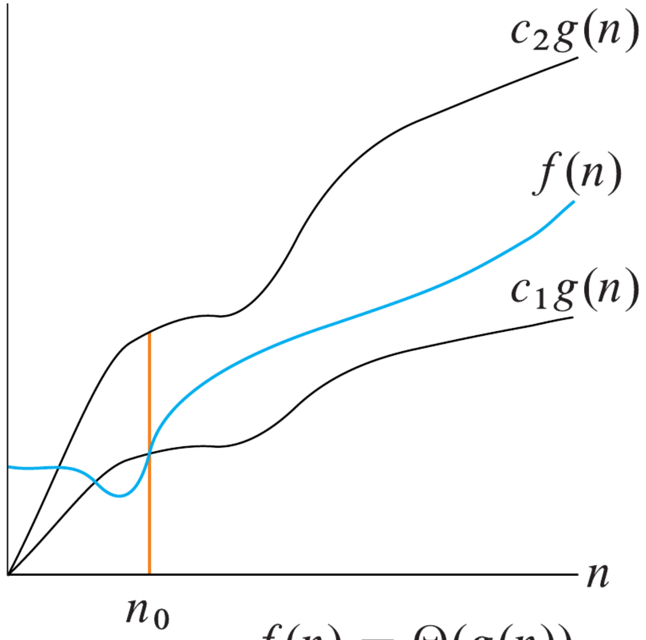
f rastie aspoň tak rýchlo ako g

$\Theta(g)$ . . . asymptotická oboustranná (těsná) mez (odhad)

Definice Pro funkci $g(n)$ je

$$\Theta(g(n)) = \{f(n) \mid \text{existují } c_1 > 0,\ c_2 > 0 \text{ a } n_0 \in \mathbb{N} \text{ tak, že pro každé } n \geq n_0 \text{ je } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)\}$$

$n_0 = \max(n_{1,0}, n_{2,0})$



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

$f = \Theta(g)$

f rastie asymptoticky rovnako ako g

Pozn.

- Dôležitý vzťah:
  - $f(n) = \Theta(g(n))$ práve keď $f(n) = O(g(n))$ a $f(n) = \Omega(g(n))$

$o(g)$ ... asymptotická ostrá (netěsná) horní mez (odhad)

Definice Pro funkci $g(n)$ je

$$o(g(n)) = \{f(n) \mid \text{pro každou } c > 0 \text{ existuje } n_0 > 0 \text{ tak, že pro každé } n \geq n_0 \text{ je } 0 \leq f(n) < cg(n)\}$$

$$f = o(g)$$

f rastie pomalšie ako g

$\omega(g)$ ... asymptotická ostrá dolní mez (odhad)

Definice Pro funkci $g(n)$ je

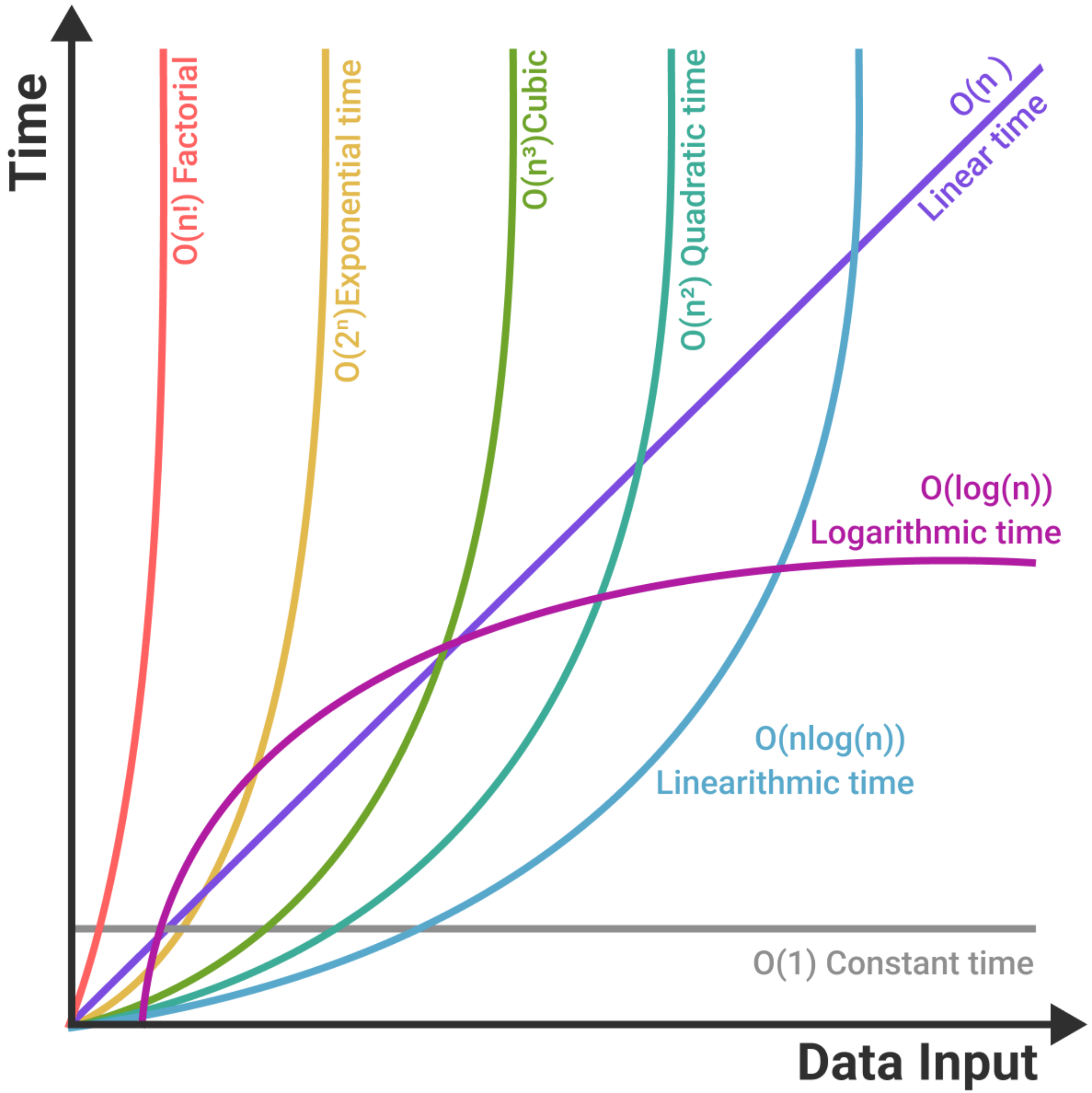
$$\omega(g(n)) = \{f(n) \mid \text{pro každou } c > 0 \text{ existuje } n_0 > 0 \text{ tak, že pro každé } n \geq n_0 \text{ je } 0 \leq cg(n) < f(n)\}$$

$$f = \omega(g)$$

f rastie rýchlejšie ako g

Top 5 Sorting Algorithms and their time complexities

Algorithm	Best Case	Average Case	Worst Case
Bubble Sort	O(N)	O(N^2)	O(N^2)
Selection Sort	O(N^2)	O(N^2)	O(N^2)
Insertion Sort	O(N)	O(N^2)	O(N^2)
Merge Sort	O(NlogN)	O(NlogN)	O(NlogN)
Quick Sort	O(NlogN)	O(NlogN)	O(N^2)


Time
O(n!) Factorial
$O(2^n)$Exponential time
$O(n^3)$Cubic
$O(n^2)$ Quadratic time
O(n)
Linear time
O(log(n))
Logarithmic time
O(nlog(n))
Linearithmic time
O(1) Constant time
Data Input

Úkol

- Dokážte, či nasledujúce výrazy platia:
  - $5n^2 + 3 = \Theta(n)$
  - $5n^2 + 3 = \Theta(n^2)$
  - $3n^3 = O(n^3)$
  - $3n^3 = o(n^3)$
  - $2n^2 + 3n - 2 = O(n!)$
  - $5\log_2 n = \Theta(\log_2 n)$
- <u>Budúci týždeň uverejním riešenia</u>