Algoritmizace

Jiří Vyskočil, Marko Genyg-Berezovskyj

2010 - 2020

Úvod

- stránky předmětu:

  https://cw.fel.cvut.cz/wiki/courses/b4b33alg/start

- cíle předmětu

  Cílem je schopnost samostatné implementace různých variant základních úloh informatiky. Hlavní témata jsou algoritmy řazení a vyhledávání a jim odpovídající datové struktury. Důraz je kladen na algoritmický aspekt úloh a efektivitu praktického řešení.

- předpoklady

  Kurs předpokládá **schopnost programování** v alespoň jednom z jazyků C/C++/Java. Součástí cvičení jsou programovací úlohy na řešení problematiky ALG. Adept musí ovládat základní datové struktury jako pole, seznam, soubor a musí být schopen manipulovat s daty v těchto strukturách.

Problémy a algoritmy

- Výpočetní problém P
  - Úkol zpracovat vstupní data IN na výstupní data OUT se zadanými vlastnostmi.
- Algoritmus A
  - Výpočetní postup řešení problému P.
  - Tedy přesný popis posloupnosti kroků, která vezme vstupní data IN a vyprodukuje výstupní data OUT dle zadaných vlastností problémem P.
- Instance problému
  - Problém s konkrétními vstupními daty potřebnými pro jeho řešení.
- Korektnost algoritmu A pro problém P
  - Algoritmus A je korektní, pokud pro každou instanci problému P vydá v **konečném** čase **správný** výstup (tedy takový, který řeší problém P).

Jak měřit algoritmy?

- Podle algoritmu vytvoříme program v programovacím jazyku a několik vybraných instancí problému.
- Algoritmy pak porovnáme podle rychlosti a paměťové náročnosti na konkrétním počítači.
- Ale co když bychom změnili počítač, nebo jen OS, nebo co kdybychom vybrali jiné instance problému, nebo kdybychom změnili programovací jazyk?
- Budou algoritmy výše popsaným způsobem stále stejně porovnatelné? .... zřejmě nikoliv ...
- → Budeme potřebovat nějakou nezávislou metodu (na programovacím jazyku, počítači, atd ...) na porovnávání algoritmů.

Růst funkcí

- Čas potřebný ke zpracování dat velikosti n, jestliže počet operací při provádění algoritmu je dán funkcí $\mathrm{T}(n)$ a provedení jedné operace trvá jednu mikrosekundu. (Připomeňme, že počet atomu ve vesmíru se odhaduje na 10^{80} a stáří na 14×10^{9} let)

$\mathrm{T}(n)/n$	20	40	60	80	100
$\log(n)$	4.3 μs	5.3 μs	5.9 μs	6.3 μs	6.6 μs
n	20 μs	40 μs	60 μs	80 μs	0.1 ms
$n\log(n)$	86 μs	0.2 ms	0.35 ms	0.5 ms	0.7 ms
n^2	0.4 ms	1.6 ms	3.6 ms	6.4 ms	10 ms
n^3	8 ms	64 ms	0.22 s	0.5 s	1 s
n^4	0.16 s	2.56 s	13 s	41 s	100 s
2^n	1 s	12.7 dní	36600 let	10^{11} let	10^{16} let
$n!$	77100 let	10^{34} let	10^{68} let	10^{105} let	10^{144} let

Asymptotické odhady

- horní asymptotický odhad (velké omikron odhad):

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

- význam:

f je shora asymptoticky ohraničená funkcí g (až na multiplikativní konstantu)

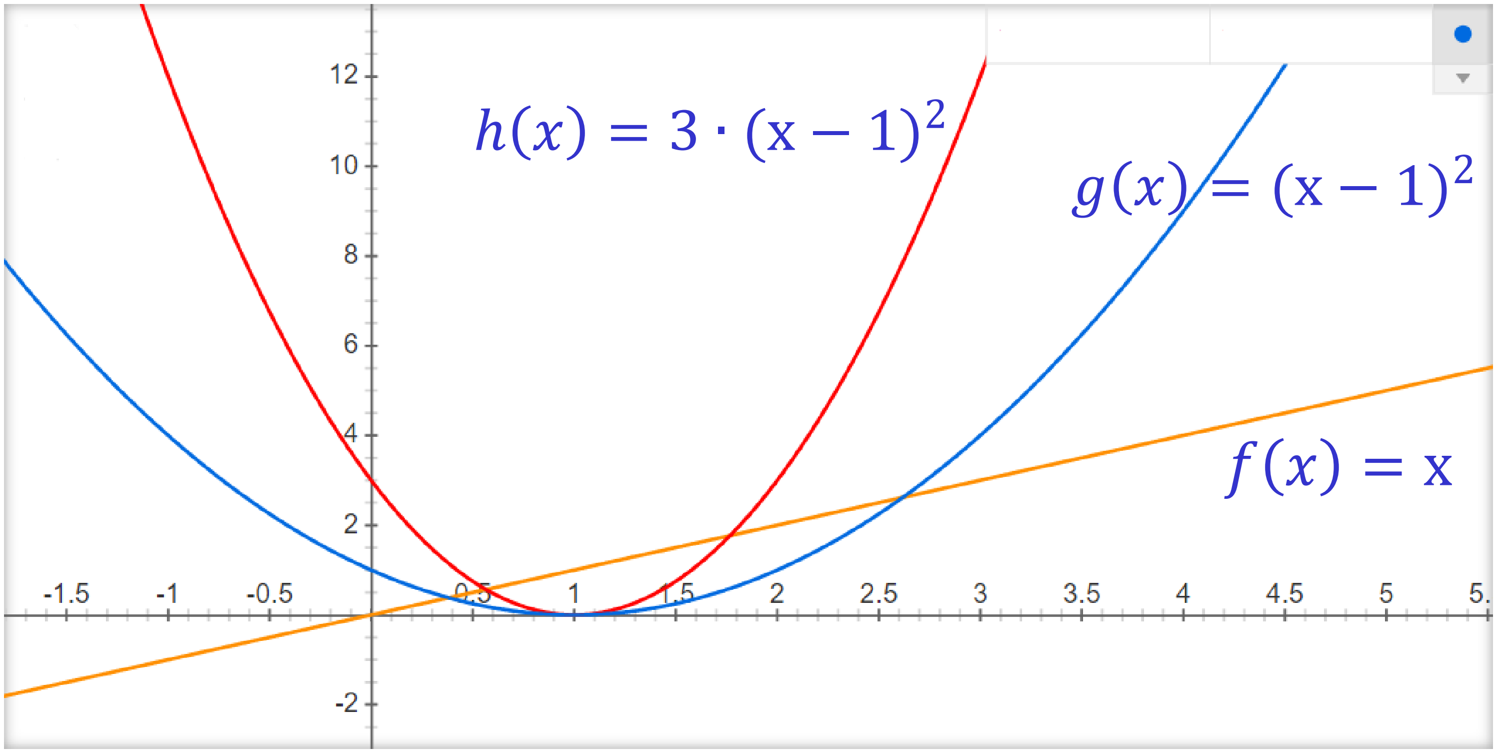
- definice:

$$(\exists c > 0)(\exists n_0)(\forall n > n_0) : f(n) \leq c \cdot g(n)$$

kde $c \in \mathbb{R}^{>0}$ $n_0, n \in \mathbb{N}$ $f, g \in \mathbb{N} \rightarrow \mathbb{R}^{\geq 0}$

Asymptotické odhady

- příklad $f(x) \in O(g(n)), \quad h(x) \in O(g(n))$



$$(\exists c > 0)(\exists n_0)(\forall n > n_0):\ f(n) \leq c \cdot g(n)$$

Asymptotické odhady

- horní asymptotický odhad pro více proměnných:

$$f(n_1, \cdots, n_k) \in O(g(n_1, \cdots, n_k))$$

- definice:

$$(\exists c > 0)(\exists n_0)(\forall n_1 > n_0) \cdots (\forall n_k > n_0):$$
$$f(n_1, \cdots, n_k) \leq c \cdot g(n_1, \cdots, n_k)$$

kde $c \in \mathbb{R}^{>0}$ $n_0, n_1, \cdots, n_k \in \mathbb{N}$ $f, g \in \mathbb{N} \to \mathbb{R}^{\geq 0}$

Asymptotické odhady

- poznámka

v literatuře se často místo

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

používá zápis

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

není to ale zcela přesné z matematického hlediska

Asymptotické odhady

- dolní asymptotický odhad (velké omega odhad):

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

- význam:

f je zdola asymptoticky ohraničená funkcí g (až na konstantu)

- definice:

$$(\exists c > 0)(\exists n_0)(\forall n > n_0): \; c \cdot g(n) \leq f(n)$$

kde $c \in \mathbb{R}^{>0}$ $\quad n_0, n \in \mathbb{N}$ $\quad f, g \in \mathbb{N} \to \mathbb{R}^{\geq 0}$

Asymptotické odhady

- optimální asymptotický odhad (velké théta odhad):

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

- význam:

f je asymptoticky ohraničená funkcí g z obou stran (až na konstantu)

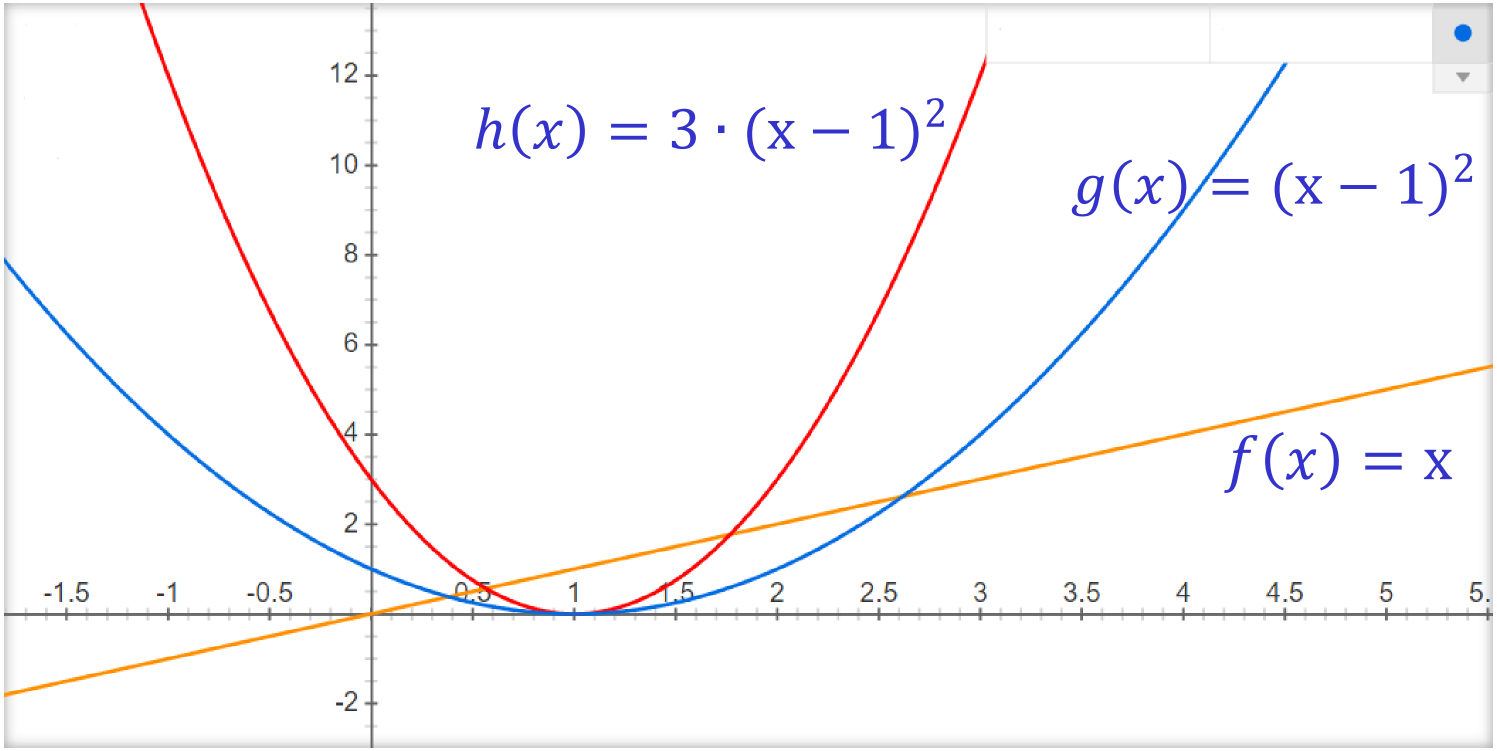
- definice: $\Theta\big(g(n)\big) \overset{\text{def}}{=} O\big(g(n)\big) \cap \Omega(g(n))$
- nebo alternativně:

$$(\exists c_1, c_2 > 0)(\exists n_0)(\forall n > n_0): c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$$

kde $c_1, c_2 \in \mathbb{R}^{>0}$ $n_0, n \in \mathbb{N}$ $f, g \in \mathbb{N} \to \mathbb{R}^{\geq 0}$

Asymptotické odhady

- příklad $g(n) \in \Theta(h(n)), \quad g(n) \notin \Theta(f(n))$



Asymptotické odhady

- příklad: Mějme dvojrozměrné pole MxN celých čísel. Jaká je asymptotická složitost problému nalezení největšího čísla v tomto poli?

- horní:
  - $O((M+N)^2)$ ✓
  - $O(\max(M,N)^2)$ ✓
  - $O(N^2)$ ✗
  - $O(M*N)$ ✓
- dolní:
  - $\Omega(1)$ ✓
  - $\Omega(M)$ ✓
  - $\Omega(M*N)$ ✓

- optimální:
  - $\Theta(M*N)$

Asymptotické odhady

- O algoritmu se složitostí $f(n)$ říkáme, že je

  logaritmický, pokud $f(n) \in \Theta(\log(n))$

  lineární, pokud $f(n) \in \Theta(n)$

  kvadratický, pokud $f(n) \in \Theta(n^2)$

  kubický, pokud $f(n) \in \Theta(n^3)$

  polynomiální, pokud $f(n) \in \Theta(n^k)$ pro $k \in \mathbb{N}$

  exponenciální, pokud $f(n) \in \Theta(k^n)$ pro $k \in \mathbb{N}$

- Poznámka: U asymptotických odhadů nemá smysl u logaritmických složitostí uvádět základ logaritmu, protože platí $\log_a(n) \in \Theta(\log_b(n))$ pro libovolná nenulová kladná a,b.

Asymptotické odhady

- Jak dokážeme $\log_a(n) \in \Theta(\log_b(n))$?

$$\boxed{\log_a n = \frac{\log_b n}{\log_b a}} = \frac{1}{\log_b a} \cdot \log_b n$$

vzoreček

konstanta

Vlastnosti asymptotických odhadů

$$
\begin{aligned}
n^m &\in O\left(n^{m'}\right) \text{ pokud } m \leq m' \\
f(n) &\in O(f(n)) \\
c \cdot O(f(n)) &= O(c \cdot f(n)) = O(f(n)) \\
O\big(O(f(n))\big) &= O\big(f(n)\big) \\
O(f(n)) + O(g(n)) &= O(\max\{f(n), g(n)\}) \\
O(f(n)) \cdot O(g(n)) &= O\big(f(n) \cdot g(n)\big) \\
O(f(n) \cdot g(n)) &= f(n) \cdot O(g(n))
\end{aligned}
$$

- Třídu složitosti polynomu určuje člen s nejvyšší mocninou:

$$\sum_{i=0}^{k} a_i \cdot n^{k-i} \in \sum_{i=0}^{k} O(n^k) = k \cdot O(n^k) = O(k \cdot n^k) = O(n^k)$$

Vlastnosti asymptotických odhadů

- Věta: Jsou-li funkce $f(n)$, $g(n)$ vždy kladné, pak pro limitu v ∞ platí
  - $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = 0$, pak $f(n) \in O(g(n))$, ale **neplatí** $f(n) \in \Theta(g(n))$
  - $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = a$, kde $0 < a < \infty$, pak $f(n) \in \Theta(g(n))$
  - $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = \infty$, pak $g(n) \in O(f(n))$, ale **neplatí** $g(n) \in \Theta(f(n))$
- Důsledek: Mějme pevně zvolené číslo $k \in \mathbb{N}$, pak platí

$$(\log(n))^k \in O(n)$$

- Důkaz lze provést pomocí L'Hopitalova pravidla.

Vlastnosti asymptotických odhadů

- Dokážeme $(\ln(n))^2 \in O(n)$

$$\lim_{x\to\infty}\frac{(\ln x)^2}{x} = \lim_{x\to\infty}\frac{\left((\ln x)^2\right)'}{x'} = \lim_{x\to\infty}\frac{2\cdot\frac{1}{x}\cdot\ln x}{1}$$

$$= \lim_{x\to\infty}\frac{2\cdot\ln x}{x} = \lim_{x\to\infty}\frac{(2\cdot\ln x)'}{x'} = \lim_{x\to\infty}\frac{2}{x} = 0$$