12.04.2018

Thursday, 12 April 2018 07:46

Wiederholung

Algorithmen

Pseudocode

Korrektheit – Invariante

Insertion sort

Analyse der Zeitkomplexität

t_j = # Durchläufe der while-Schleife in Abh. von j.

Best case: $t_j = 1$ falls Folge sortiert ist.

Worst case: $t_j = j$ falls Folge umgekehrt sortiert

$$\sum_{j=2}^{n} t_j \quad \begin{cases} \sum_{j=2}^{n} 1 = n-1 & \text{Best case} \\ \sum_{j=2}^{n} j = \frac{n(n+1)}{2} - 1 & \end{cases}$$

Worst case

$$\sum_{j=1}^{n} = \frac{n(n+1)}{2}$$

Worst case

$$n\,(c_1 + c_2 + c_4 + c_8) - c_2 - c_4 - c_8$$

$$+ \underbrace{\left(\frac{n(n+1)}{2} - 1\right)}_{= \frac{n^2 + n}{2} - 1} (c_5 + c_6 + c_7) - (n-1)(c_6 + c_7)$$

$$n^2 \left(\frac{c_5}{2} + \frac{c_6}{2} + \frac{c_7}{2}\right)$$

$$+\, n \left(c_1 + c_2 + c_4 + \frac{c_5}{2} - \frac{c_6}{2} - \frac{c_7}{2} + c_8\right)$$

$$-\, (c_2 + c_4 + c_5 + c_8)$$

quadratisch

Best case : linear, optimal

$\Theta(f)$ wurde auf Folie definiert und "fast" gezeigt

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

$$6n^3 \overset{?}{\in} \Theta(n^2)$$

$$\lim_{n \to \infty} \frac{6n^3}{n^2} = \infty$$

$$n \to \infty \quad n^2$$

Angenommen, es gäbe ein passendes n_0, c_2, dann wäre

$$6n^3 \leq c_2 n^2 \quad \forall n \geq n_0$$

$$6n \leq c_2 \quad \forall n \geq n_0$$

falsch für hinreichend großes n, nämlich für $n > \frac{c_2}{6}$

$O(g)$ definiert auf Folie.

$$1000\, n^2 = O(n^2 \log n)$$

$$n^2 \log n = \Omega(1000 n^2)$$

$f \in O(g)$ – f wächst* nicht schneller als g

$f \in \Omega(g)$ – f wächst* nicht langsamer als g

$f \in o(g)$ – f wächst* langsamer als g

* asymptotisch

Theorem ① $f \in \Theta(g) \Leftrightarrow$

$f \in O(g) \wedge f \in \Omega(g)$

② $f \in O(g) \Leftrightarrow g \in \Omega(f)$

$f \,\Theta\, g \Leftrightarrow f \in \Theta(g)$

die anderen Relationen analog

transitiv O, Ω, Θ, o?

reflexiv O, Ω, Θ

symmetrisch Θ

$\Theta(1)$ ~~konstant~~

$\Theta(n)$ linear

$\Theta(n(\log n)^m)$ quasilinear

$\Theta(n^3)$ kubisch

$\Theta(n^k)$ polynomiell

$\Theta(2^n)$ exponentiell