Helmut Schauer
Educational Engineering Lab
Department for Informatics
University of Zurich

Komplexitätstheorie

Helmut Schauer
Educational Engineering Lab
Department for Informatics
University of Zurich

Eigenschaften von Algorithmen

- **deterministisch - nichtdeterministisch**
- **sequentiell - parallel**
- **endlich - unendlich**
- **reversibel - irreversibel**

Helmut Schauer
Educational Engineering Lab
Department for Informatics
University of Zurich

Ordnung einer Funktion O(f)

$f \in O(g) \Leftrightarrow$ Ex c, n_0: $c>0$:(All n: $n \geq n_0$: $f(n) \leq c*g(n)$)

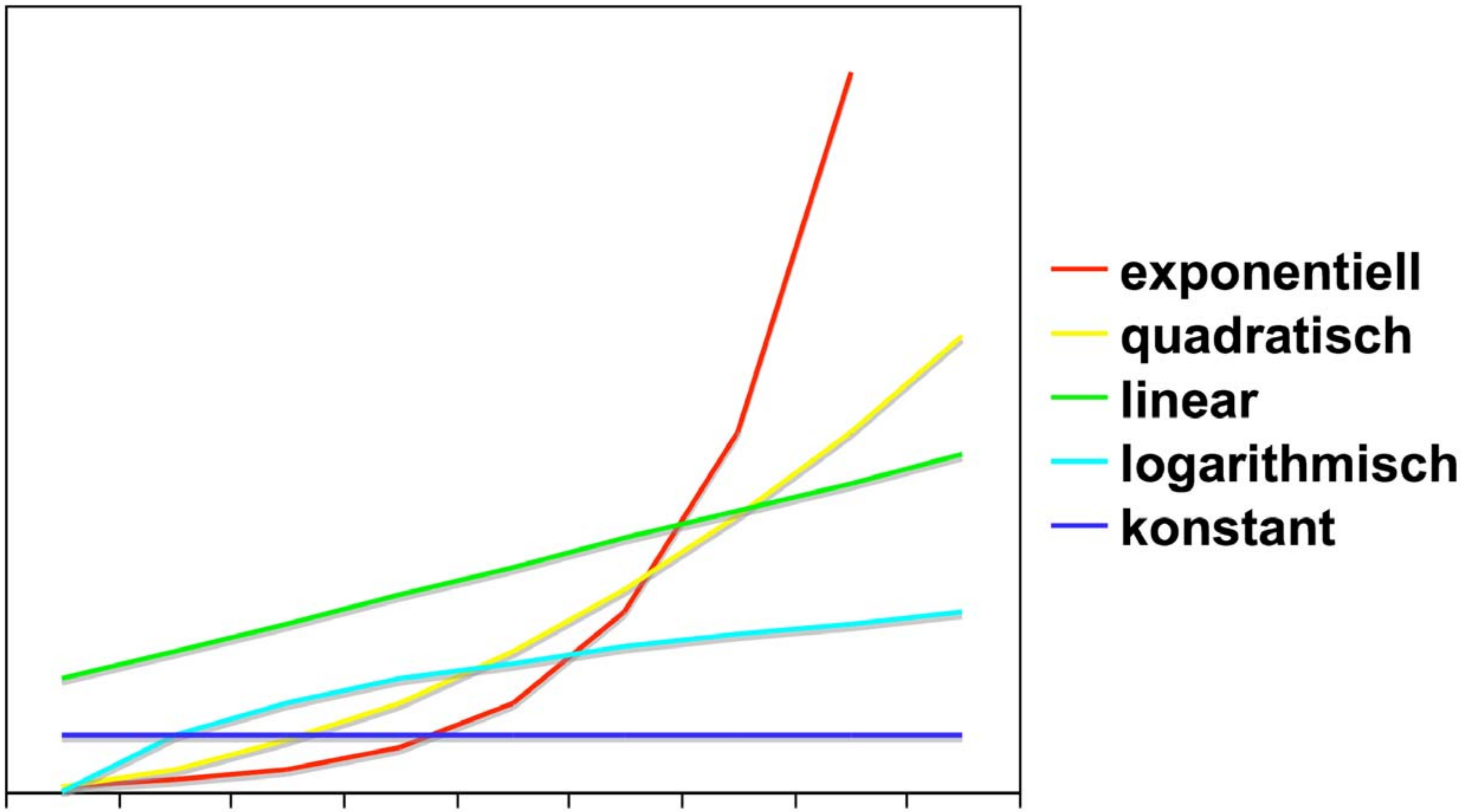
oder

$$f \in O(g) \Leftrightarrow \lim_{n \to \infty} f(n)/g(n) = c$$

$f \in O(g)$ ... „f ist von der Ordnung g“

Helmut Schauer
Educational Engineering Lab
Department for Informatics
University of Zurich

Typische Ordnungen von Funktionen



Helmut Schauer
Educational Engineering Lab
Department for Informatics
University of Zurich

Bezeichnungen

$O(1)$	**konstant**
$O(\log n)$	**logarithmisch**
$O(n)$	**linear**
$O(n^2)$	**quadratisch**
$O(n^k)$	**polynomial**
$O(e^n)$	**exponentiell**

Helmut Schauer
Educational Engineering Lab
Department for Informatics
University of Zurich

n	**ld n**	**n ld n**	**n^2**	**2^n**	**n!**
10	3	33	100	1024	$3*10^6$
20	4	86	400	10^6	$2*10^{18}$
100	7	664	10’000	10^{31}	10^{161}
1000	10	10’000	10^6		
10000	13	130’000	10^8		

zum Vergleich: es gibt etwa 10^{79} Protonen im Universum

Helmut Schauer
Educational Engineering Lab
Department for Informatics
University of Zurich

n	ld n	n ld n	n^2	2^n	n!
10	3 µs	33 µs	100 µs	1 ms	3 s
20	4 µs	86 µs	400 µs	1 s	10^5 Jahre
100	7 µs	664 µs	10 ms	10^{17} Jahre	
1000	10 µs	10 ms	1 s		
10000	13 µs	130 ms	100 s		

zum Vergleich: der Urknall war vor etwa 15 Milliarden Jahren

Helmut Schauer
Educational Engineering Lab
Department for Informatics
University of Zurich

Regeln

$O(c*f(n)) = O(f(n))$
$O(f(n)+g(n)) = \max(O(f(n)),O(g(n)))$

$O(f(n)) \leq O(g(n)) \Leftrightarrow f(n) \in O(g(n))$
$O(f(n)) = O(g(n)) \Leftrightarrow (O(f(n)) \leq O(g(n)))$ and $(O(g(n)) \leq O(f(n)))$
$O(f(n)) < O(g(n)) \Leftrightarrow (O(f(n)) \leq O(g(n)))$ and $(O(g(n)) \neq O(f(n)))$

Helmut Schauer
Educational Engineering Lab
Department for Informatics
University of Zurich

Beispiele

$O(2*n-1) = O(n)$

$O(n*(n+1)/2) = O(n^2)$

$O(ld\ n) = O(\log n)$

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

$O(n*\log n) < O(n^2)$

$O(\log n) < O(n^{1/2})$

Helmut Schauer
Educational Engineering Lab
Department for Informatics
University of Zurich

Beispiel: Binäres Suchen

$$A(n) = 1 + A(n/2)$$
$$A(1) = 1$$

$$A(n) = 1 + \mathrm{ld}\ n \;\Rightarrow\; O(A(n)) = O(\log n)$$

Helmut Schauer
Educational Engineering Lab
Department for Informatics
University of Zurich

P- und NP-Probleme

P-Probleme sind mit polynomialem Aufwand lösbar
NP-Probleme sind nicht mit polynomialem Aufwand lösbar

NP steht für “nichtdeterministisch polynomial”

Ob P=NP oder P≠NP ist ist ungelöst!

Helmut Schauer
Educational Engineering Lab
Department for Informatics
University of Zurich

NP-vollständige Probleme

Alle NP-vollständigen Probleme können mit polynomialem Aufwand aufeinander abgebildet werden.

Falls ein einziges der NP-vollständigen Probleme mit polynomialem Aufwand gelöst werden kann gilt P=NP!

Helmut Schauer
Educational Engineering Lab
Department for Informatics
University of Zurich

Beispiele für NP-vollständige Probleme:

Traveling Salesperson
Knapsack
Scheduling
Bin-Packing
Graph Coloring
Satisfiability

Helmut Schauer
Educational Engineering Lab
Department for Informatics
University of Zurich

Nicht berechenbare Probleme:

Es gibt Aufgaben die nicht algorithmisch lösbar sind.

zB: Halteproblem: Es gibt keinen Algorithmus der entscheidet ob ein beliebiges Programm terminiert!