

Komplexität von Algorithmen

Big-O-Notation (asymptotische obere Schranke)

$f(n)$ ist von $\mathcal{O}(g(n))$ genau dann, wenn $n_0 \in \mathbb{N}$ und $c \in \mathbb{R} +$ existieren mit $f(n) \leq c \cdot g(n)$ für alle $n \leq n_0$.

Beispiele

Komplexität	Bedeutung	Algorithmus
$\mathcal{O}(1)$	konstant	Element aus einem Array holen
$\mathcal{O}(\log n)$	logarithmisch	Binäre Suche
$\mathcal{O}(\sqrt{n})$	wie Wurzel	Naiver Primzahltest
$\mathcal{O}(n)$	linear	Lineare Suche
$\mathcal{O}(n \log n)$	superlinear	Mergesort, Heapsort
$\mathcal{O}(n^2)$	quadratisch	Selectionsort
$\mathcal{O}(n^m)$	polynomiell	Einfache Algorithmen
$\mathcal{O}(2^n)$	exponentiell	
$\mathcal{O}(n!)$	faktoriell	Problem des Handlungsreisenden