

O-Notation



Generell ist es sehr sinnvoll, den Rechenzeitaufwand eines Algorithmus in Abhängigkeit von n (Datenumfang oder andere entscheidende Problemgröße) abzuschätzen.

Hierzu eignet sich die von Paul Bachmann 1894 eingeführte O-Notation an:

$g(n) = O(f(n))$ für $n \in \mathbb{N}$, falls $\exists M, n_0 \in \mathbb{N}$, so daß gilt

$$|g(n)| \leq M |f(n)| \quad \forall n \geq n_0.$$

Einige Beispiele:

$O(1)$	konstanter Aufwand, unabhängig von n
$O(n)$	linearer Aufwand (z.B. Einlesen von n Zahlen)
$O(n \ln n)$	Aufwand guter Sortierverfahren (z.B. Quicksort)
$O(n^2)$	quadratischer Aufwand
$O(n^k)$	polynomialer Aufwand (bei festem k)
$O(2^n)$	exponentieller Aufwand
$O(n!)$	Bestimmung aller Permutationen von n Elementen

Die O-Notation hilft insbesondere bei der Beurteilung, ob ein Algorithmus für großes n noch geeignet ist bzw. erlaubt einen Effizienz-Vergleich zwischen verschiedenen Algorithmen für große n .



Copyright © 1999 [Andreas Borchert](#), in HTML konvertiert am 29.06.1999