

Die folgende Übersicht fasst die relevanten Informationen zur Landaunotation für Laufzeiten (auch Zeitkomplexitäten genannt) von Algorithmen zusammen. Mit Laufzeit ist dabei die Anzahl der Operationen, nicht die tatsächliche Zeit gemeint.



### Laufzeiten / betrachtete Fälle

- worst case - Anzahl benötigte Operationen im schlechtesten Fall
- best case - s.o., jedoch im besten Fall
- average case - arithmetisches Mittel aller möglichen Laufzeiten

Achtung: Zeitkomplexität  $T$  immer in Abhängigkeit der Länge der Eingabe ( $n$ ) angeben!  $\Rightarrow T(n)$ ...

### Landau-Notation

Zur Abschätzung der Laufzeit.

$T(n) \in O(n)$  bedeutet, die Laufzeit wächst in Abhängigkeit von  $n$  nicht wesentlich schneller als die Funktion  $g(n) = n^2$

Es gibt weitere Landau-Symbole, die wir aber nicht behandeln.



### Abschätzung und Vereinfachung

Nach dem Ermitteln einer Laufzeitfunktion erfolgt eine Abschätzung bzw. Einordnung in die Komplexitätsklassen. Dabei fallen alle den Funktionsverlauf nicht wesentlich beeinflussende Elemente weg.

Beispiel: Aus  $T(n) = 6n^2 - 3n + 999$  wird  $T(n) \in O(n^2)$

| Komplexitäts-klasse<br>(ansteigend) | in Worten                                | ggf. Beispiel                 |
|-------------------------------------|------------------------------------------|-------------------------------|
| 1                                   | konstant (es gibt eine feste Obergrenze) | ist eine Binärzahl gerade?    |
| $\log(n)$                           | logarithmisch                            | binäre Suche                  |
| $n$                                 | linear                                   | lineare Suche                 |
| $n \log(n)$                         | „super-linear“                           | Merge-/Heapsort               |
| $n^2$                               | quadratisch                              | Bubble- Insert- Selectionsort |
| $n^m$                               | polynomiell                              |                               |
| $2^n$                               | exponentiell                             |                               |
| $n!$                                | faktoriell                               | Traveling Salesman            |



### Effiziente Algorithmen

Ein Algorithmus gilt dann als effizient, wenn er maximal polynomielle Zeitkomplexität besitzt.