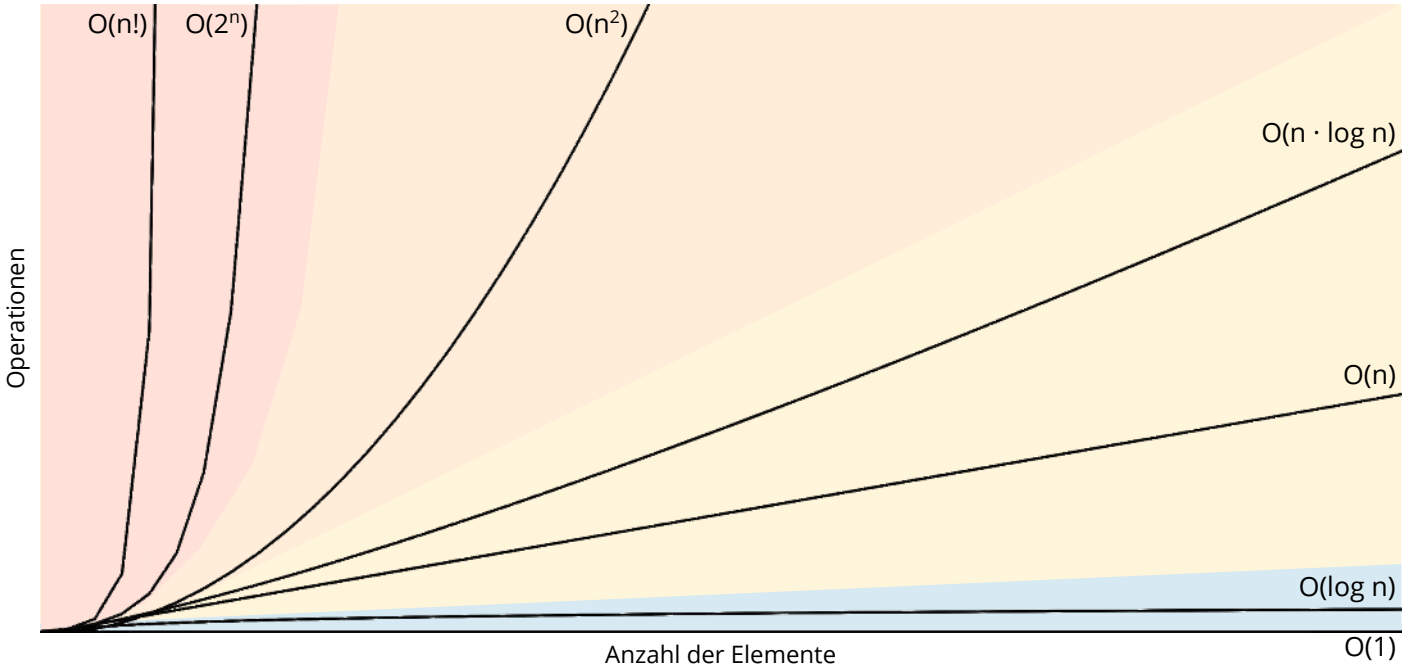
O-Notation – Cheat Sheet

www.happycoders.eu



Komplexitätsklassen erklärt

Komplexitätsklasse	Beschreibung	Beispiel	
Konstanter Aufwand O(1)	Die Anzahl der Operationen bleibt gleich, unabhängig von der Anzahl der Elemente.	Zugriff auf ein spezifisches Element eines Arrays.	Ausgezeichnet
Logarithmischer Aufwand O(log n)	Die Anzahl der Operationen steigt um einen konstanten Betrag, wenn sich die Anzahl der Elemente verdoppelt.	Mit binärer Suche ein Element in einem *sortierten* Array finden.	Ausgezeichnet
Linearer Aufwand O(n)	Die Anzahl der Operationen wächst linear mit der Anzahl der Elemente *n*. Wenn sich *n* verdoppelt, verdoppelt sich auch die Anzahl der Operationen.	Ein Element in einem *unsortierten* Array finden.	Mittelmäßig
Quasilinearer Aufwand O(n · log n)	Die Anzahl der Operationen wächst etwas schneller als linear, da die lineare Komponente mit einer logarithmischen multipliziert wird.	Effiziente Sortierverfahren wie Quicksort, Mergesort, Heapsort.	Mittelmäßig
Quadratischer Aufwand O(n²)	Die Anzahl der Operationen wächst linear mit dem Quadrat der Anzahl der Elemente *n*. Wenn sich *n* verdoppelt, vervierfacht sich die Anzahl der Operationen.	Einfache Sortierverfahren wie Insertion Sort und Selection Sort.	Schlecht
Exponentieller Aufwand O(2ⁿ)	Die Anzahl der Operationen wächst exponentiell mit der Anzahl der Elemente. Sie verdoppelt sich für jedes zusätzliche Element.	Rekursive Fibonacci-Methode.	Sehr schlecht
Faktorieller Aufwand O(n!)	Die Anzahl der Operationen wächst linear mit der Fakultät der Anzahl der Elemente *n*, die das Produkt aller Zahlen bis (und einschließlich) *n* ist.	Brute-Force-Lösung für das Traveling-Salesman-Problem.	Sehr schlecht