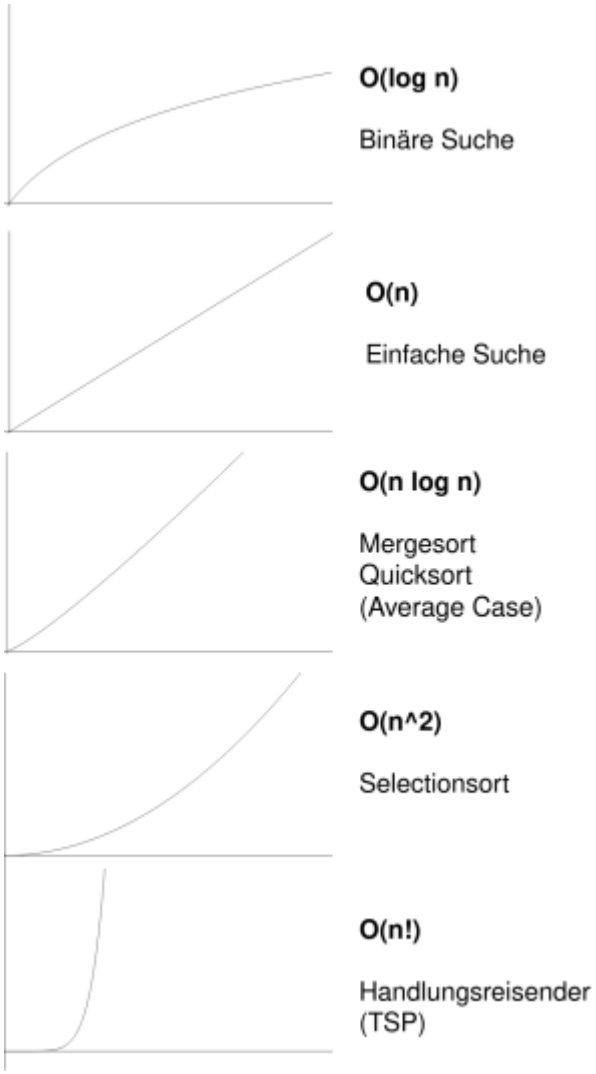
Aufwandsabschätzung im Detail



Im Abschnitt zur binären Suche haben wir uns bereits einige Gedanken zur Aufwandsabschätzung gemacht.

Um ein Gefühl dafür zu bekommen, was die gängigsten Laufzeitcharakteristiken bedeuten, können die folgenden Beispiele dienen:

	10 Elemente	**100 Elemente**	**1000 Elemente**
O(log n)	0,15 Sekunden	0,3 Sekunden	0,5 Sekunden
O(n)	0,5 Sekunden	5 Sekunden	50 Sekunden
O(n log n)	1,6 Sekunden	33 Sekunden	490 Sekunden
$O(n^2)$	5 Sekunden	8 Minuten	14 Stunden
O(n!)	2,1 Tage	$1{,}4*10^{149}$ Jahre	$0{,}6*10^{2559}$ Jahre

Hypothetisch wurden für diese sehr grobe Berechnung eine Bearbeitungsgeschwindigkeit von ca. 20 Operationen je Sekunde zugrunde gelegt, was natürlich sehr viel langsamer ist, als ein Computer real arbeitet.

Es ist aber wichtig zu verstehen, dass bei Problemen der Kategorie $O(n^2)$ oder gar O(n!) keine Rolle spielt: Wenn die Anzahl der Elemente wächst, **wächst der Aufwand schneller als jede**

Rechenleistung das zu kompensieren vermag[1)].

Quicksort

Eine Besonderheit des Quicksort-Algorithmus ist, dass er Aufwand von der Wahl des Pivotelement abhängt, das hast du vielleicht bei deinen Übungen bereits bemerkt: Wenn man das Element stets sehr ungünstig wählt, gewinnt man bei Aufteilen des Problem kaum etwas, die nach der Partitionierung größte zu sortierende Menge ist im schlechtesten Fall in jedem Rekursionsschritt nur ein Element kleiner als zuvor, wobei die kleinste Menge immer leer ist.

1)

Bei den Beispielen haben wir ja gerade mal 1000 Elemente betrachtet, das sind ja eigentlich lächerlich wenige…