

Lernweg Datenbanken



Verwendung: Du kannst die Informationen zu den Lernwegabschnitten in dein Notizprogramm übernehmen oder ausdrucken und in dein Heft kleben, so dass nach jedem Schritt Raum zur Selbstreflexion und für eigene Notizen bleibt.

1: Sortieren und Aufwandsbeurteilung

Sortialgorithmen sind relativ aufwändig - an dieser Stelle kann man sich also auch damit beschäftigen, wie man den Aufwand eines Algorithmus beurteilt und vergleicht. Zum Einstieg betrachten wir die [binäre Suche](#), sie es ermöglicht auf sortierten Daten effizient bestimmte Elemente zu finden.

Anschließend betrachten wir die eine [vereinfachte Variante der sogenannten "O-Notation"](#), mit der man den Aufwand eines Algorithmus klassifizieren kann.

2: Sortierverfahren

Java bietet mit dem Comparable Interface eine Möglichkeit, auch eigene Datentypen (Klassen) sortierbar zu machen. Um die Problemstellung "Sortieren" einzuordnen, schauen wir uns das hier an: [Warum sortieren - das Comparable Interface](#).

Jetzt können wir in die Sortierverfahren einsteigen, zunächst betrachten (und implementieren) wir verschiedene sogenannte vergleichsbasierte Sortierverfahren:

- [Bubble Sort](#)
- [Selection Sort](#)
- [Insertion Sort](#)
- [Mergesort](#)
- [Quicksort](#)

Mergesort und Quicksort sind rekursive Verfahren, die das ["Teile und herrsche Prinzip"](#), das du bereits aus der Rekursion kennst, verwenden.

Von Bedeutung ist die Aufwandsbeurteilung der Sortialgorithmen und eine Eigenschaft, die man ["stabil"](#) nennt, die manche Sortialgorithmen haben - und andere nicht.

3: Ergänzung: Nicht vergleichsbasierte Sortierverfahren

Unter besonderen Rahmenbedingungen kann man Sortierverfahren finden, die nicht vergleichsbasiert funktionieren, z.B. [Radixsort](#).

From:
<https://info-bw.de/> -

Permanent link:
<https://info-bw.de/faecher:informatik:oberstufe:algorithmen:sorting:lernweg:start>

Last update: **13.05.2025 05:58**

