

Sortieren

- Sortieren bedeutet, eine Menge von Daten nach einem bestimmten Kriterium zu ordnen
 - Sehr häufige benötigt bei der Verarbeitung von Daten

Aufgabe 1

- Sortiere die folgende Zahlenfolge aufsteigend:

17 12 3 19 8 21

- Menschen können eine (kurze) Zahlenfolge mehr oder weniger intuitiv in die richtige Reihenfolge bringen
- Ein Computer(programm) ist hier eingeschränkt
 - Welche Grundoperationen sind für das Sortieren wichtig?
- Die meisten Sortierverfahren basieren darauf, dass
 - paarweise die Werte von Elementen verglichen werden
 - ggf. die Position von Elementen getauscht wird

- Acht Freiwillige, die sich in einer Reihe aufstellen
- Ein:e weitere:r Freiwillige:r
 - Aufgabe: Sortieren nach Schuhgröße

- Sehr einfaches Sortierverfahren
- Ablauf:
 - Vergleiche der Reihe nach alle Elemente paarweise
 - Tausche, falls ein Paar die Sortierreihenfolge verletzt
 - Wiederhole, bis alle Elemente sortiert sind

Beispiel:

- Die Zahlfolge 17, 12, 3, 19, 8, 21 soll aufsteigend sortiert werden

17 12 3 19 8 21 Vergleiche 17 mit 12, tausche

12 17 3 19 8 21 Vergleiche 17 mit 3, tausche

...

- Führe Bubblesort für die angegebenen Zahlen “von Hand” (also mit Stift und Papier) durch
 - Woher weiß man, dass man fertig ist?
 - Warum nennt sich das Verfahren Bubblesort?

- Implementiere Bubblesort
 - Vervollständige die Methoden `sortiere()` und `tausche()`

- Wir betrachten ein Array mit n Elementen
 - Im ersten Durchlauf werden maximal $(n - 1)$ Vergleiche durchgeführt
 - Im zweiten Durchlauf werden maximal $(n - 2)$ Vergleiche durchgeführt
 - ...
- Insgesamt also $\frac{n(n-1)}{2} = \frac{1}{2}n^2 - \frac{1}{2}n$ Vertauschungen
 - Entscheidend ist hier der Teilterm n^2
 - Konstante Vorfaktoren und kleinere Potenzen sind vernachlässigbar
- Laufzeit Bubblesort: $O(n^2)$

- Die Laufzeit von Bubblesort wächst quadratisch mit der Zahl der zu sortierenden Elemente
 - Bei einer Verdoppelung der Array-Größe ist also eine Vervierfachung der Laufzeit zu erwarten
 - Kann man das messen?

Aufgabe 4

- Erweitere Dein Programm aus Aufgabe 3 um eine Zeitmessung
 - Verwende die Java-Methode `long System.nanoTime()` (liefert zurück, wie viele Nanosekunden seit einem bestimmten, festen Zeitpunkt vergangen sind)
- Bestimme die Laufzeit für verschiedene Größen des zu sortierenden Arrays
 - Führe alle Messungen ggf. mehrfach durch und bestimme den Mittelwert