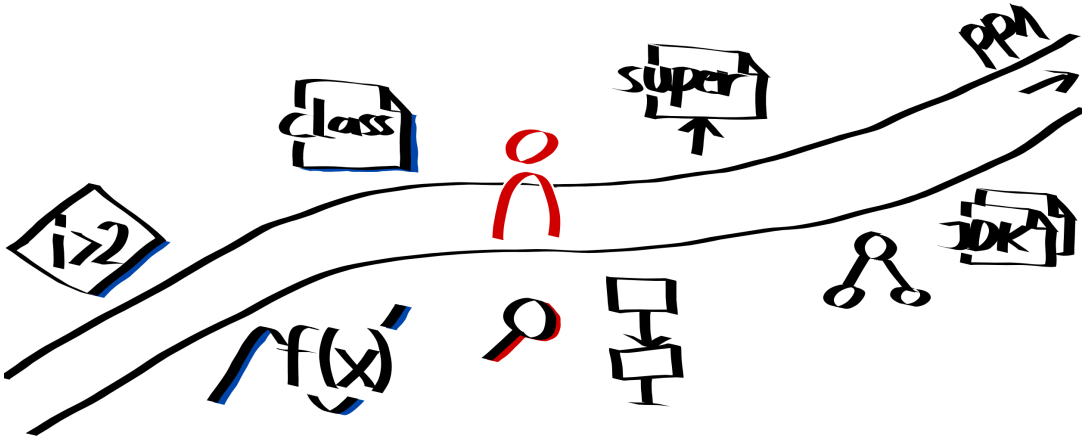
Kapitel 4: Suchen & Sortieren

VL 15: Sortieren durch Einfügen

Wintersemester 2023/24

Stand 27. November 2023, 18:43 Uhr

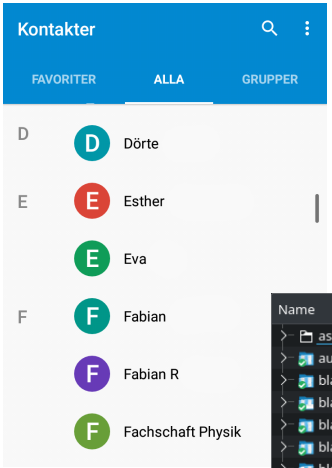
Wo stehen wir gerade?

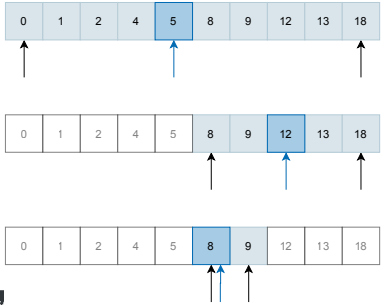


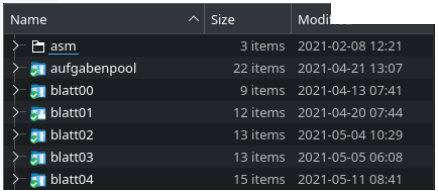
Vorlesungsfeedback

Warum Daten sortieren?

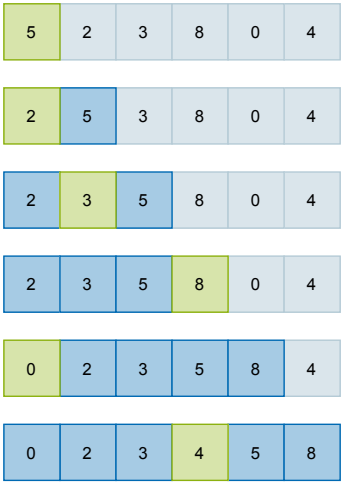
Warum Daten sortieren?







Erste Idee



Insertion Sort[1]

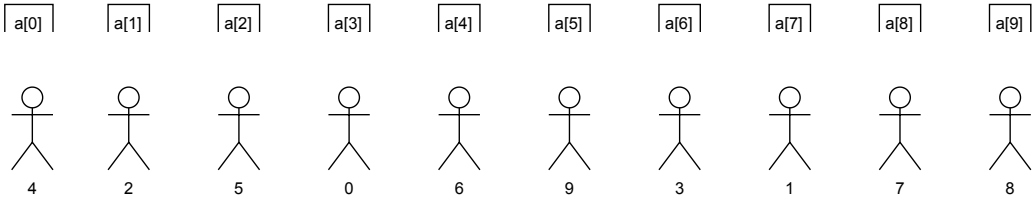
Heinrich Heine Universität Düsseldorf

- Idee: Bewege Element an Index i so lange nach links, bis korrekte Position erreicht; beginne bei $i = 0$
  - $\rightarrow$ nach und nach entsteht links ein sortiertes Teil-Array

```java
private static void insertionSort(int[] numbers) {
    for(int currentIndex = 1; currentIndex < numbers.length; currentIndex++) {
        int currentNumber = numbers[currentIndex];
        int newPosition = currentIndex;
        while(newPosition > 0 && numbers[newPosition - 1] > currentNumber) {
            numbers[newPosition] = numbers[newPosition - 1];
            newPosition--;
        }
        numbers[newPosition] = currentNumber;
    }
}
```

[1] Sortieren durch Einfügen

Insertion Sort als Tanz



Insert-sort with Romanian folk dance:

`https://www.youtube.com/watch?v=ROalU379l3U`

Studis sortieren

Heinrich Heine
Universität Düsseldorf

Aufgabe

Gegeben: Array mit Studierende (Name + Matrikelnummer).
Ziel: Nach Matrikelnummer sortieren.

Formen sortieren

Heinrich Heine
Universität Düsseldorf

Aufgabe

Gegeben: Array mit Bällen.
Ziel: Nach Flächeninhalt sortieren.

Wie können wir beliebige Objekte sortieren?

Heinrich Heine Universität Düsseldorf

- Voraussetzung: Objekte brauchen Sortierschlüssel
- Lösung: Polymorphie

```
1 public interface Sortable {
2     int getSortKey();
3 }
```

  - Sortierbare Objekte implementieren Interface
  - Sortier-Methode arbeitet mit allen Arrays, die nur Objekte enthalten, die Interface implementieren

→ weniger redundanter Code

→ Sortiermethode funktioniert auch mit in Zukunft erstellten Datentypen, die Interface implementieren (Open-Closed-Prinzip →Propra)

Insertion Sort mit Sortable

```java
private static void insertionSort(Sortable[] objects) {
    for(int currentIndex = 1; currentIndex < objects.length; currentIndex++) {
        Sortable currentObject = objects[currentIndex];
        int newPosition = currentIndex;
        while(newPosition > 0 && objects[newPosition - 1].getSortKey() >
          ↪   currentObject.getSortKey()) {
            objects[newPosition] = objects[newPosition - 1];
            newPosition--;
        }
        objects[newPosition] = currentObject;
    }
}
```

Benutzt man „in Echt“ auch ein Interface?

Benutzt man „in Echt“ auch ein Interface?

Heinrich Heine Universität Düsseldorf

Ja!

- Interface `Comparable<T>` im JDK[2]
- funktioniert ein bisschen anders, aber gleicher Grundgedanke

[2] `https://docs.oracle.com/javase/8/docs/api/java/lang/Comparable.html`

Wie schnell ist Insertion Sort? (theoretisch)

Wie schnell ist Insertion Sort? (theoretisch)

Heinrich Heine
Universität Düsseldorf

- schlechtester Fall: Liste falschherum sortiert
- Wie viele Elemente müssen beim „Einfügen“ verschoben werden?

Wie schnell ist Insertion Sort? (theoretisch)

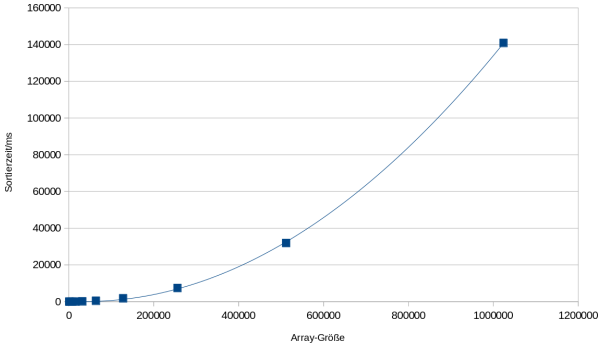
Heinrich Heine Universität Düsseldorf

- schlechtester Fall: Liste falschherum sortiert
- Wie viele Elemente müssen beim „Einfügen“ verschoben werden?
  - beim 1. Element: 0
  - beim 2. Element: 1
  - beim 3. Element: 2
  - ...
- Durchschnitt: $\frac{n}{2}$ Verschiebungen
- insgesamt: $\frac{n^2}{2}$ Verschiebungen $\Rightarrow$ quadratische Laufzeit im schlechtesten Fall
- im durchschnittlichen Fall auch quadratische Laufzeit $\rightarrow$AlDat

Wie schnell ist Insertion Sort? (empirisch)

- durchschnittliche Sortierdauer steigt quadratisch mit Array-Größe

→ egal, wie schnell Computer ist: Verdopplung der Array-Größe ⇒ Suchdauer vervierfacht

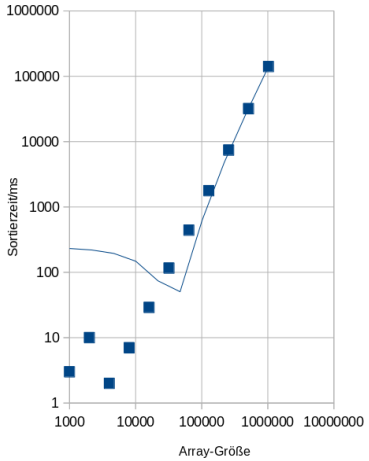


Wie schnell ist Insertion Sort? (Log-Log-Plot[3])

- Hypothese: an^2
- $\log(an^2) = \log(a) + 2\log(n)$

$\Rightarrow$ Steigung im Log-Log-Plot gibt Exponenten

- Steigung in diesem Diagramm: ~ 2

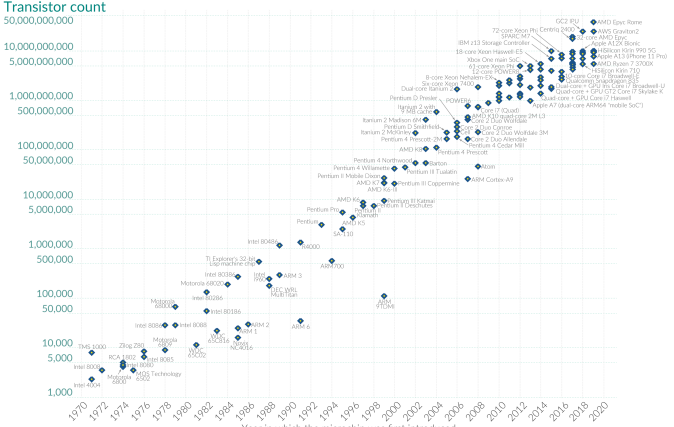


[3] https://en.wikipedia.org/wiki/Log–log_plot

Kann man nicht einfach schnellere Computer bauen?

Heinrich Heine
Universität Düsseldorf

Kann man nicht einfach schnellere Computer bauen?



https://ourworldindata.org/uploads/2020/11/Transistor-Count-over-time.png, https://creativecommons.org/licenses/by/4.0/

Lernziele

Heinrich Heine
Universität Düsseldorf

Sie können am Ende der Woche ...

- einen Sortieralgorithmus **implementieren**.

Wichtige Begriffe

Heinrich Heine
Universität Düsseldorf

Insertion Sort quadratische Laufzeit