

# Aufbau des Moduls

- ▶ Vorlesung: Mo 10–12 CAP3 H3, Mi 16–18 CAP3 H3
- ▶ Übung Mo–Mi
- ▶ Globalübung: Do 16–18 CAP3 H3

## Arbeitsaufwand

- ▶ 8 LP entsprechen  $8 \times 30h = 240h$  pro Semester
- ▶ Wochen des Semesters: ca 14
- ▶ D.h ca. 17h pro Woche
- ▶ 8h für Vorlesung, Übung und Globalübung
- ▶ ca. 9h Eigenarbeit pro Woche!

# Übungsbetrieb

- ▶ Theorie- und Programmieraufgaben
- ▶ Freitag: Ausgabe der Aufgaben
- ▶ Donnerstag 14:00 Uhr: Abgabe der Lösungen im Schrein (Theorie), per Ilearn (Praktisch)
- ▶ Donnerstag 16–18 Uhr: Besprechung der Hausaufgaben (Globalübung)
- ▶ Montag–Mittwoch: Rückgabe der Hausaufgaben (Übung)
- ▶ Erste Serie 15. 4.; Erste Übung: 18. 4.

# Prüfungsleistung

Klausur jeweils im 1. und 2. Prüfungszeitraum

## Klausurzulassung

- ▶ 40% der Theorie Punkte pro Serie, 2 Joker
- ▶ 4 von 6 Programmieraufgaben korrekt bearbeiten

# Zuständigkeiten

- ▶ *Inhaltliche* Fragen zu Vorlesung, Übungsaufgaben oder Korrektur: Übungsgruppenleiter
- ▶ Probleme technischer Art oder andere Anmerkungen:
  - ▶ Hausaufgaben: Kim
  - ▶ Programmierung: Felix
  - ▶ Präsenzaufgaben: Malin
- ▶ Webseite: [algo.informatik.uni-kiel.de](http://algo.informatik.uni-kiel.de)
  - ▶ Lehre
  - ▶ Algorithmen und Datenstrukturen

# Termine

## Java-Einführung

In der ersten Globalübung am 14.4.

## Probeklausur

7.6. CAP3 H2

# Motivation

## Algorithmus

Ein Algorithmus ist eine eindeutige Handlungsvorschrift zur Lösung eines Problems oder einer Klasse von Problemen. Algorithmen bestehen aus endlich vielen, wohldefinierten Einzelschritten. [...]

Bei der Problemlösung wird eine bestimmte Eingabe in eine bestimmte Ausgabe überführt.

## Datenstruktur

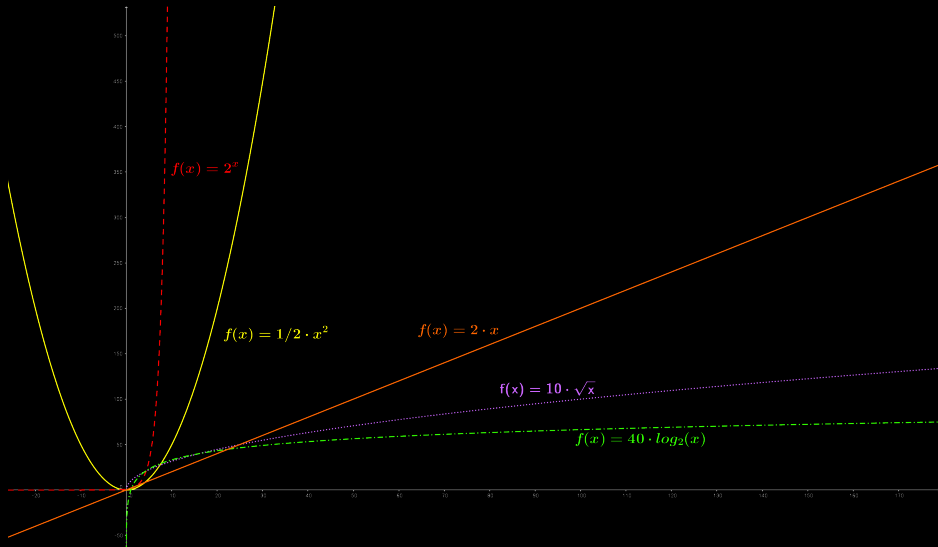
In der Informatik und Softwaretechnik ist eine Datenstruktur ein Objekt zur Speicherung und Organisation von Daten. [...]

Datenstrukturen sind nicht nur durch die enthaltenen Daten charakterisiert, sondern vor allem durch die Operationen auf diesen Daten, die Zugriff und Verwaltung ermöglichen und realisieren.

# Beispiel: Sortieren

Demonstration: Insertion-Sort vs. Quicksort

# Verschiedene Laufzeitklassen I





## Verschiedene Laufzeitklassen II

Laufzeit bei Eingabegröße $x$	Maximale Problemgröße in 100s	Max. Problemgröße in 100s, 10x so schnell
$2^x$	6	10
$\frac{1}{2}x^2$	14	44
$2x$	50	500
$10\sqrt{x}$	100	10 000
$10 \log(x)$	1 024	> 1 267 650 Trilliarden

# Testprogramm

- ▶ Herunterladen des Testprogramms
- ▶ Herunterladen der Vorlage
- ▶ Entpacken des Testprogramms und der Vorlage
- ▶ info.cfg anpassen

## Ausführung des Testprogramms:

```
python test.py <Name> <Pfad>
```

```
Beispiel: python test.py Sort ../Sort
```

# Inhalt

- ▶ O-Notation
- ▶ Analyse von Algorithmen (Laufzeit, Speicherplatz) am Beispiel von Sortierverfahren
- ▶ Algorithmische Methoden
  - ▶ Binäre Suche
  - ▶ Rekursion
  - ▶ Dynamische Programmierung
  - ▶ Divide and Conquer
  - ▶ Backtracking
- ▶ Datenstrukturen: Listen, Suchbäume, Hashtabellen
- ▶ Algorithmen auf Graphen

# Litteratur

- ▶ Norbert Blum: Algorithmen und Datenstrukturen: eine anwendungsorientierte Einführung, Oldenbourg 2004.
- ▶ Thomas H. Cormen, Charles E. Leiserson, und Ronald L. Rivest, Clifford Stein: Introduction to Algorithms, Boston: MIT Press, 2001.
- ▶ Donald E. Knuth: The Art of Computer Programming. Vol. 1: Fundamental Algorithms, 3rd ed., Addison-Wesley 1997. Vol. 3: Sorting and Searching, 2nd ed., Addison-Wesley 1998.
- ▶ Sven Oliver Krumke, Hartmut Noltemeier: Graphentheoretische Konzepte und Algorithmen, Teubner 2005.

# Litteratur

- ▶ Harald Reß und Günter Viebeck: Datenstrukturen und Algorithmen: objektorientiertes Programmieren in C++, Hanser 2000.
- ▶ Robert Sedgewick: Algorithms in Java, Parts 1-4, 3rd ed., Addison-Wesley, 2002.
- ▶ Mark Allen Weiss: Data Structures and Algorithm Analysis in Java, 2nd ed., Addison-Wesley, 2007.
- ▶ Mehlhorn und Sanders: Algorithms and Data Structures: The Basic Toolbox, Springer