

# Algorithmen und Programmierung

## Kapitel 1 Einführung

<http://www.tu-ilmenau.de/telematik/aup>



## Wer sind wir?

- ❑ Fachgebiet „Telematik/Rechnernetze“
  - ❑ Prof. Dr.-Ing. Günter Schäfer  
[guenter.schaefer@tu-ilmenau.de](mailto:guenter.schaefer@tu-ilmenau.de)  
Raum Zusebau 3042
  - ❑ M.Sc. David Schatz  
[david.schatz@tu-ilmenau.de](mailto:david.schatz@tu-ilmenau.de)  
Raum Zusebau 3077
- ❑ Web-Adressen:
  - ❑ Fachgebiet: <http://www.tu-ilmenau.de/telematik>
- ❑ Sprechstunde Prof. Schäfer: Montags, 13:00 - 14:00 Uhr



- ❑ Allgemein: Architekturen und Protokolle für Kommunikationssysteme
  - ❑ Struktur, Entwurf, Leistungsbewertung, Implementierung, ...
  - ❑ Für alle Arten von Kommunikation: Rechnernetze, Sprachkommunikation, Daten und Multimedia-Inhalte, technische Kommunikation (Steuerung)
  - ❑ Besonderer Fokus auf Sicherheitsaspekte
- ❑ Spezifische Aspekte:
  - ❑ Sicherheitsanforderungen von Diensten
    - Authentizität, Integrität, Vertraulichkeit von Partnerinstanzen und ausgetauschten Daten
  - ❑ Sicherheitsaspekte von Protokollmechanismen
    - Welche sicherheitsrelevanten Seiteneffekte implizieren einzelne/mehrere Mechanismen?
  - ❑ Schutz von Kommunikationsinfrastrukturen
    - Hauptaspekt: Gewährleistung der Verfügbarkeit von Systemen und angebotenen Diensten



- ❑ Sommersemester:
  - ❑ VL *Advanced Networking Technologies*
  - ❑ VL *Telematik I* (Inf., Ing.Inf., WI, Math)
  - ❑ VL *Schutz von Kommunikationsinfrastrukturen*
  - ❑ Hauptseminar/Proseminar *Telematik*
  - ❑ Projektseminar *Simulative Bewertung von Internet-Protokollfunktionen*
- ❑ Wintersemester:
  - ❑ VL *Telematik 2/Leistungsbewertung*
  - ❑ VL *Network Security*
  - ❑ VL *Netzalgorithmen*
  - ❑ Hauptseminar/Proseminar *Telematik*
  - ❑ Projektseminar *Simulative Bewertung von Internet-Protokollfunktionen*



1. Einführung: Historie, Grundbegriffe
2. Algorithmische Grundkonzepte: Sprachen, Grammatiken, Datentypen, Terme
3. Java: Einführung, Programmiergrundlagen
4. Algorithmenparadigmen
5. Algorithmenmodelle
6. Eigenschaften von Algorithmen
7. Ausgewählte Algorithmen
8. Abstrakte Datentypen
9. Objektorientierung
10. Datenstrukturen



Saake, G.; Sattler, K.:  
**Algorithmen und Datenstrukturen:  
Eine Einführung mit Java.**

5. Auflage, dpunkt-Verlag, 2013  
576 Seiten, 44.90 €

Website:

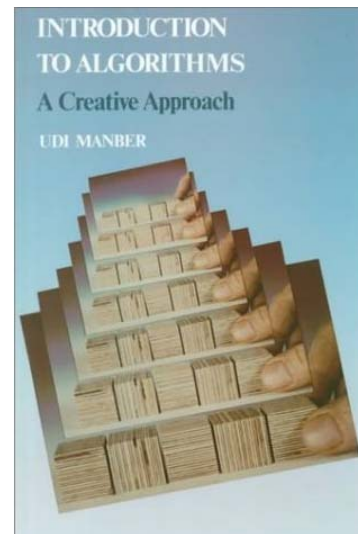
[http://wwwiti.cs.uni-magdeburg.de/iti\\_db/algoj](http://wwwiti.cs.uni-magdeburg.de/iti_db/algoj)



Manber, Udi:  
**Introduction to Algorithms –  
A Creative Approach.**

Addison Wesley, 1989  
478 Seiten, 77.50 €

(in Bibliothek bzw. gebraucht  
erhältlich)



- M.T. Goodrich, R. Tamassia.  
*Data Structures and Algorithms in Java.*  
3. Auflage, Wiley, 2003.
- R. Sedgewick.  
*Algorithmen in Java, Teil 1-4.*  
3. Auflage, Pearson Studium, München, 2002.
- T. Ottmann, P. Widmayer.  
*Algorithmen und Datenstrukturen.*  
Spektrum Akademischer Verlag, 2000.
- H. Mössenböck.  
*Sprechen Sie Java?*  
2. Auflage, dpunkt-Verlag, 2003



- Alle Infos & Folienkopien unter  
<http://www.tu-ilmenau.de/telematik/aup>
- Veranstaltung besteht aus:
  - 1.5 Vorlesungen pro Woche (da 1 Termin 14-tägig)
  - 1 Kleingruppenübung pro Woche
  - Selbständige Bearbeitung der Übungsblätter (wichtig!)
    - Theorie- & Programmieraufgaben
    - Abgabe jeweils Montag um 11:00 Uhr
    - Durch Abgabe der Übungsblätter können Bonuspunkte zur Anrechnung auf die beiden Vorbereitungsklausuren (s.u.) erworben werden
  - Vorbereitungsklausuren:
    - Zwei Vorbereitungsklausuren im Semester (02.12.2019 und 27.01.2020)
    - Mindestens eine davon muss für die Zulassung zur Modulprüfung bestanden werden
    - Werden beide bestanden, können zudem Bonuspunkte zur Anrechnung auf die Modulprüfung erworben werden
    - Anmeldung (eine Anmeldung für beide) mit Thoska+ bis zum 25.11.2019
  - Modulprüfung: Klausur im Prüfungszeitraum



- Wer hat im Abi einen Informatik-Kurs belegt?
- Wer besitzt einen Computer oder hat regelmäßigen Zugang?
- Wer hat schon mal programmiert?
- Wer kennt Java (nicht den Kaffee oder die Insel)?
- . . . oder Pascal, BASIC, C, etc.?



---

Jetzt geht's los!



# Teil I

## Einführung



## „ehuedfn“

Vorbemerkungen

Grundbegriffe

Algorithmen

Eigenschaften von Algorithmen

Historischer Überblick



- **Programmieren** als wichtige Tätigkeit von \*-Informatikern
  - Anwendungssysteme, Abläufe, Steuerungen / Geräte, . . .
- „Strukturiertes Denken“: systematische und strukturierte Analyse und Lösung von Aufgaben
  - Eindeutige Beschreibung von durch Rechner bearbeitbaren Aufgaben (**Algorithmen**)



- Kunstwort aus den 60ern
  - Informatik → Information + Technik *oder*
  - Informatik → Information + Mathematik
- Beabsichtigt: Gegensatz zur amerikanischen *Computer Science*: nicht nur auf Computer beschränkt
- Bereiche:
  - Theoretische Informatik: mathematische Modelle, theoretische Konzepte (Logik, Spezifikation, Komplexität)
  - Praktische Informatik: Techniken der Programmierung und Realisierung von Softwaresystemen
  - Technische Informatik: Struktur und Aufbau von Computern
  - Angewandte Informatik: Anwendungen von Informatiksystemen
  - „Bindestrich-Informatiken“





- Informatik hat zentral zu tun mit:
  - Systematischer Verarbeitung von Informationen
  - Maschinen, die diese Verarbeitung automatisch leisten ( → Computer)
- Hier: maschinenunabhängige Darstellung

Die „systematische Verarbeitung“ wird durch den Begriff **Algorithmus** präzisiert, Information durch den Begriff **Daten**



- Allgemein:

Ein Algorithmus ist eine eindeutige Beschreibung eines in mehreren Schritten durchgeführten (Bearbeitungs-)Vorganges.

- In der Informatik speziell: Berechnungsvorgänge statt Bearbeitungsvorgänge, Schwerpunkt auf *Ausführbarkeit* durch (abstrakte) Maschinen.

Ein *Prozessor* führt einen Prozess (Arbeitsvorgang) auf Basis einer eindeutig interpretierbaren Beschreibung (dem Algorithmus) aus.



- Kochrezepte
  - Pizza aufwärmen
  - Seeteufel mit Kräuterkruste auf Lauch
  
- Bedienungsanleitungen
  - Suchen eines Telefonbucheintrags
  - Herausgeben von Wechselgeld
  - Sortieren von Spielkarten
  
- Berechnungsvorschriften
  - Schriftliches Addieren
  - Berechnung der Fakultät ( $x! = x \cdot (x - 1) \cdot \dots \cdot 3 \cdot 2 \cdot 1$ )
  - Berechnung des größten gemeinsamen Teilers



- Notation für Beschreibung
- Ausdrucksfähigkeit
- Berechenbarkeit
- Korrektheit / Genauigkeit / Eindeutigkeit
- Zeitbedarf / Geschwindigkeit
  
- In dieser Vorlesung:

**Algorithmen für Computer**





$$0! = 1$$
$$x! = x \cdot (x - 1)!$$

```
X = 20; y = 1;
while (x > 1) {
    y = y * x;
    x = x - 1;
}
return y;
```



- ❑ Verschiedene Notationen = gleiche Ausdrucksfähigkeit?
- ❑ Wahl der Programmiersprache – universelle Programmiersprache?
- ❑ Aber, man vergleiche z. B.
  - ❑ Bienensprache vs. Sprache zur Wegfindungsprogrammierung für Roboter



- ❑ Hundesprache vs. menschliche Sprache



- Kann man „alles“ programmieren (berechnen)?

Antwort: **Nein!**

- Nichtentscheidbare Probleme:
  - Halteproblem: Entscheide, ob ein (beliebiger) gegebener Algorithmus terminiert
  - Semantische Eigenschaften von Algorithmen, z. B.:
    - Berechnen zwei Algorithmen dieselbe Funktion?
    - Ist ein gegebener Algorithmus korrekt, d. h. berechnet er die gegebene (gewünschte) Funktion?
  - Die Beweise hierfür werden in Vorlesungen zur theoretischen Informatik behandelt



- Algorithmen (Programme) sollen sich wie beabsichtigt verhalten  
→ **Korrektheit**
- Beispiele für Programmfehler  
([de.wikipedia.org/wiki/Programmfehler](https://de.wikipedia.org/wiki/Programmfehler))

- 1962: Verlust der Venus-Sonde Mariner 1 durch fehlenden Bindestrich in einem Fortran-Programm (80 Mill. Dollar)
- 1996: Zerstörung der Ariane 5-Rakete kurz nach dem Start durch übernommenen und nicht korrekt spezifizierten Programmcode
- 1999: Verlust der Mars-Sonde Climate Orbiter durch falsches Maßsystem (Yard statt Meter) bei Programmierung



- Wie aufwändig ist ein gegebener Algorithmus?
- Abschätzung des Aufwands:
  - Unabhängig von konkreter Hardwareleistung (Moore'sches Gesetz: „Komplexität von integrierten Schaltungen / Rechenleistung verdoppelt sich alle 18 Monate“)
  - In Abhängigkeit von der Problemgröße
  - Für den schlechtesten Fall (worst case) und im Mittel
- Beispiel:

Operation	Aufwand	$n = 2$	$n = 2^{10}$	$n = 2^{20}$
sequenzielle Suche	$n$	2	1024	1048576
Baum-Suche	$\log n$	1	10	20



- 300 v. Chr.: Euklids Algorithmus zur Bestimmung des **ggT**, (7. Buch der Elemente):

$$\text{ggT}(300, 200) = 100$$

- 800 n. Chr.: Muhammed ibn Musa abu Djafar al-Choresmi: Aufgabensammlung für Kaufleute und Testamentsvollstrecker (lat.: Liber Algorithmi, Kunstwort aus dem Namen und griechisch „arithmos“ für Zahl)
- 1574: Adam Rieses Rechenbuch
- 1614: Logarithmentafeln (30 Jahre für Berechnung!)
- 1703: Binäres Zahlensystem (Leibnitz)
- 1931: Gödels Unvollständigkeitssatz
- 1936: Church'sche These



Fragen?

