Algorithmen und Berechnungskomplexität I

Heiko Röglin
Institut für Informatik I

Wintersemester 2010/11

Organisatorisches

Vorlesung

- Montags 11:15 - 12:45 Uhr (AVZ III / HS 1)
- Mittwochs 11:15 - 12:45 Uhr (AVZ III / HS 1)

Dozent

- Heiko Röglin
- Professor für theoretische Informatik
- Institut für Informatik, Abteilung I
- Arbeitsgebiet: Entwurf und Analyse von Algorithmen

Übungen

- wöchentliche Übungen

Mo	13 (c.t.) - 15	AVZ III / A7a
Mi	13 (c.t.) - 15	AVZ III / A7a
Do	11 (c.t.) - 13	AVZ III / A7a
Do	13 (c.t.) - 15	AVZ III / A301
Fr	11 (c.t.) - 13	AVZ III / A7a
Fr	13 (c.t.) - 15	AVZ III / A301

- Anmeldung: Geben Sie den Anmeldezettel spätestens Mittwoch (13.10.) in der Vorlesung ab.

Übungsblätter und Übungsschein

- jeden Mittwoch gibt es ein **Übungsblatt**
  - Abgabe: am Mittwoch darauf in der Vorlesung
  - Besprechung: in der Woche nach der Abgabe
  - Gruppenarbeit: Abgabe in Gruppen mit max. 3 Studierenden
- Voraussetzung für den **Übungsschein**:
  - insgesamt 50% der möglichen Punkte auf den Übungszetteln
  - aktive Teilnahme an den Übungen
    (das heißt insbesondere Anwesenheit in den Übungen)
- **Probeklausur** am 15. Dezember 2010
  - zur besseren Selbsteinschätzung
  - Möglichkeit, Zusatzpunkte für die Übungen zu erhalten

Klausur

- Übungsschein ist Zulassungsvoraussetzung
- Datum: Wahrscheinlich in der Woche vom 7. bis zum 11. Februar (der genaue Termin ist abhängig von der Verfügbarkeit von Räumen und wird bald mitgeteilt)
- Ähnlichkeit mit Übungsaufgaben und Probeklausur
- zweite Klausur: Ende März
- vor der zweiten Klausur ist ein Ferientutorium geplant

Fragen

Bei Fragen

- Sprechstunde: Dienstags 10-11 Uhr oder nach der Vorlesung
- E-Mail: `roeglin@uni-bonn.de`
- Büro: Brühler Straße 7, Raum 3.052
  (Bei verschlossener Tür 0228 / 73 - 4326 anrufen.)
- Weitere Informationen auf der Webseite:
  `http://www.roeglin.org/teaching/WS2010/AuBI.html`
- Feedback zur Veranstaltung ist erwünscht.

Literatur

Jede Vorlesung basiert auf einem Kapitel in einem Buch/Skript, anhand dessen sie nachgearbeitet werden kann.

Literatur

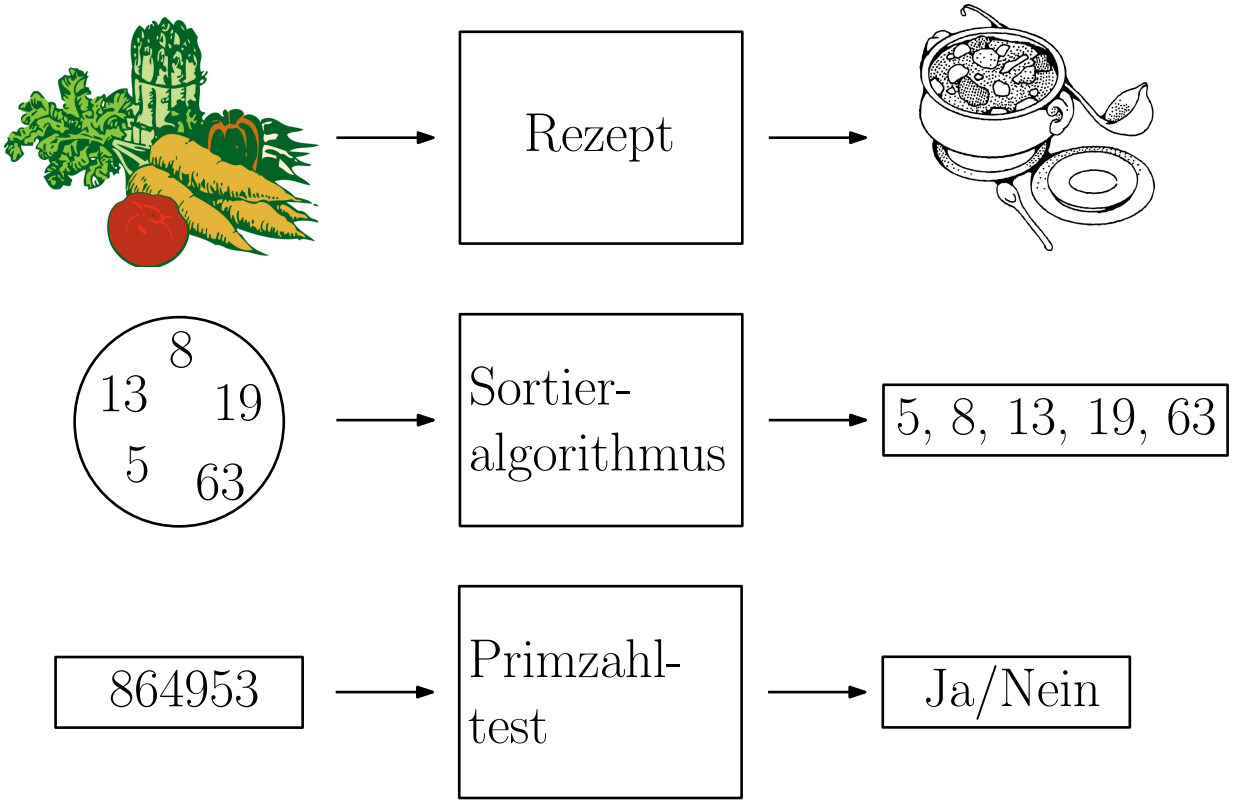
- Introduction to Algorithms
  Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein
  ISBN-13: 978-0262533058, Dritte Auflage, MIT Press
- Skript zur Vorlesung „Berechenbarkeit und Komplexität“
  Berthold Vöcking, RWTH Aachen, Wintersemester 2009/10
- Skript zur Vorlesung „DAP II“
  Ingo Wegener, Universität Dortmund, Sommersemester 2007
- Material zum Kurs „Introduction to Algorithms“ am MIT

  `http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/`

  `6-046j-introduction-to-algorithms-sma-5503-fall-2005/`

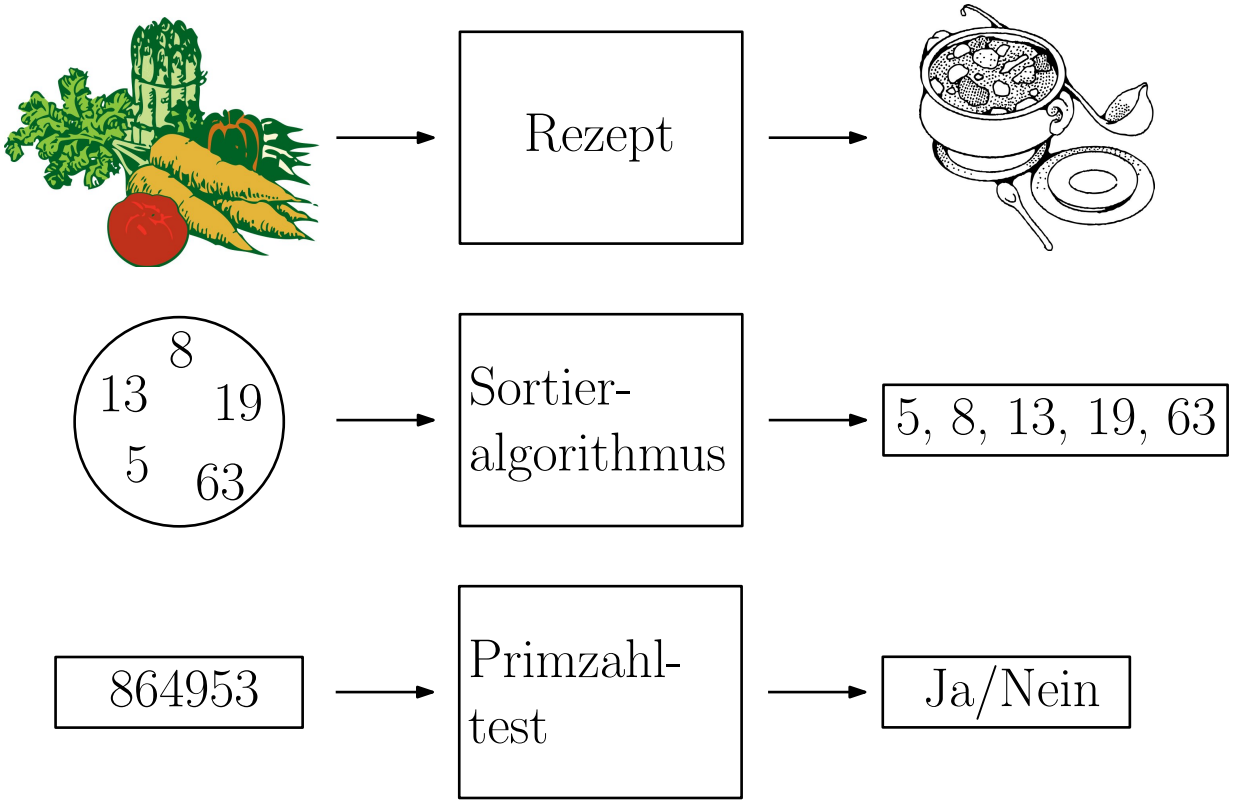
Überblick – Algorithmen

Ein **Algorithmus** ist ein formal definiertes Verfahren, das eine Eingabe in eine Ausgabe transformiert (ein „Kochrezept“).



Überblick – Algorithmen

Ein **Algorithmus** ist ein formal definiertes Verfahren, das eine Eingabe in eine Ausgabe transformiert (ein „Kochrezept“).

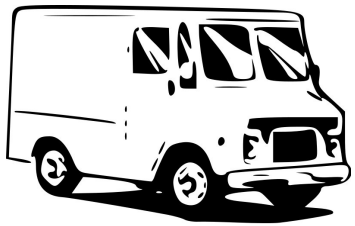


Formal definiert $\cong$ genau genug beschrieben, um direkt in Java/C++ implementiert werden zu können

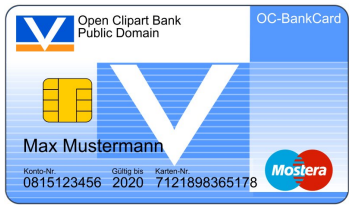
Überblick – Algorithmen

Algorithmen werden überall gebraucht:

Navigationsgerät:
Was ist der kürzeste Weg von A nach B?



Paketdienst:
In welcher Reihenfolge sollen Kunden beliefert werden?



(Online) Banking:
Wie werden Informationen ver- und entschlüsselt?

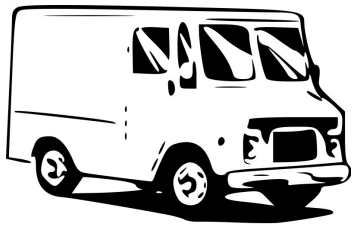
Google

Suchmaschinen:
Wie findet man Informationen in einer großen Datenbank?

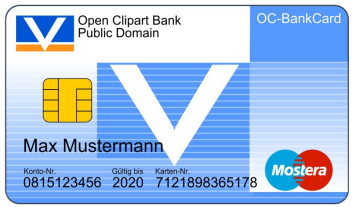
Überblick – Algorithmen

Algorithmen werden überall gebraucht:

Navigationsgerät:
Was ist der kürzeste Weg von A nach B?



Paketdienst:
In welcher Reihenfolge sollen Kunden beliefert werden?



(Online) Banking:
Wie werden Informationen ver- und entschlüsselt?

Google

Suchmaschinen:
Wie findet man Informationen in einer großen Datenbank?

Trotz schneller Rechner: Effizienz ist wichtig!

Überblick – Berechnungskomplexität

Wo liegen die Grenzen?

```
i:= 10;
while (i > 1) i++;
```

VS.

```
i:= 10;
while (i > 1) i--;
```

Halteproblem:

Terminiert Programm nach endlich vielen Schritten?

Überblick – Berechnungskomplexität

Wo liegen die Grenzen?

```
i:= 10;
while (i > 1) i++;
```

VS.

```
i:= 10;
while (i > 1) i--;
```

Halteproblem:

Terminiert Programm nach endlich vielen Schritten? Nicht entscheidbar!

Überblick – Berechnungskomplexität

Wo liegen die Grenzen?

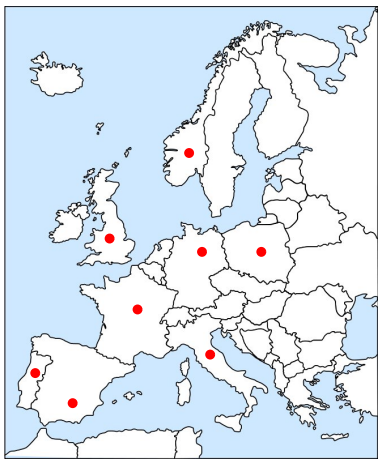
```
i:= 10;
while (i > 1) i++;
```

VS.

```
i:= 10;
while (i > 1) i--;
```

Halteproblem:

Terminiert Programm nach endlich vielen Schritten? Nicht entscheidbar!



Problem des Handlungsreisenden (TSP):

Finde kürzeste Tour durch alle Städte.

Überblick – Berechnungskomplexität

Wo liegen die Grenzen?

```
i:= 10;
while (i > 1) i++;
```

VS.

```
i:= 10;
while (i > 1) i--;
```

Halteproblem:

Terminiert Programm nach endlich vielen Schritten? Nicht entscheidbar!

Problem des Handlungsreisenden (TSP):

Finde kürzeste Tour durch alle Städte.

Überblick – Berechnungskomplexität

Wo liegen die Grenzen?

```
i:= 10;
while (i > 1) i++;
```

VS.

```
i:= 10;
while (i > 1) i--;
```

Halteproblem:

Terminiert Programm nach endlich vielen Schritten? Nicht entscheidbar!

Problem des Handlungsreisenden (TSP):

Finde kürzeste Tour durch alle Städte.

Vermutlich nicht effizient lösbar!

Ausblick

- Wintersemester 2010/11:
  Algorithmen und Berechnungskomplexität I (V4+Ü2)
  - grundlegende Datenstrukturen
  - grundlegende Graphalgorithmen
  - Sortieralgorithmen
  - allgemeine Methoden zum Entwurf von Algorithmen
- Sommersemester 2011:
  Algorithmen und Berechnungskomplexität II (V4+Ü2)
  - Welche Problem sind nicht berechenbar?
  - Welche Probleme sind nicht effizient berechenbar?
  - Mächtigkeit von Automaten und Sprachen

Lernziele

- Kenntnis grundlegender Datenstrukturen, Methoden und Konzepte
- Fähigkeit, ein gegebenes Problem zu analysieren:
  - Einordnung der Schwierigkeit des Problems
  - Auswahl geeigneter Datenstrukturen und Methoden zur Lösung des Problems
- Entwurf eines Algorithmus zur Lösung eines gegebenen Problems
- Analyse des entwickelten Algorithmus