Grundlagen der Algorithmen und Datenstrukturen

Christian Scheideler + Helmut Seidl

SS 2009

Organisatorisches

- Vorlesung:

  Di 12:00 – 13:30 MI HS 1
  Do 12:00 – 12:45 MI HS 1

- Übungen: 34 Alternativen (max. 15 Teiln.)
  4 englische
- Übungsleitung: Sylvia Pott, Vivian Prinz
  uebungsleitung-gad@mailseidl.in.tum.de
- Anmeldung: Grundstudiumstool

Organisatorisches

- Webseite: www2.in.tum.de/hp/?GAD09
- Literatur (u.a.):

  Kurt Mehlhorn und Peter Sanders.
  Algorithms and Data Structures –
  The Basic Toolbox
  Springer Verlag, Berlin, 2008.

Webseite enthält Vorlesungsfolien und Übungsblätter

Organisatorisches

- Übungsblätter:
  Jede Woche ein Übungsblatt.
  Ausgabe: jeden Dienstag auf der Kurswebseite
  Abgabe: jeden Dienstag vor der Vorlesung

- Eine Klausur:

  voraussichtlich am 25. Juli (Samstag)
  Anmeldung über TUM Online

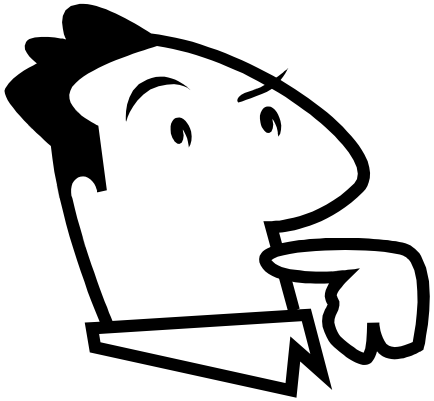
- Endnote:

  Klausur + Bonus
  Bonus: Bestehen + 50% der Übungspunkte: 0,3

Einführung

Thema: Algorithmen und Datenstrukturen

Theorie?



Muss ich Programme schreiben?

- Was ist ein Algorithmus?
- Was ist eine Datenstruktur?
- Einführende Beispiele

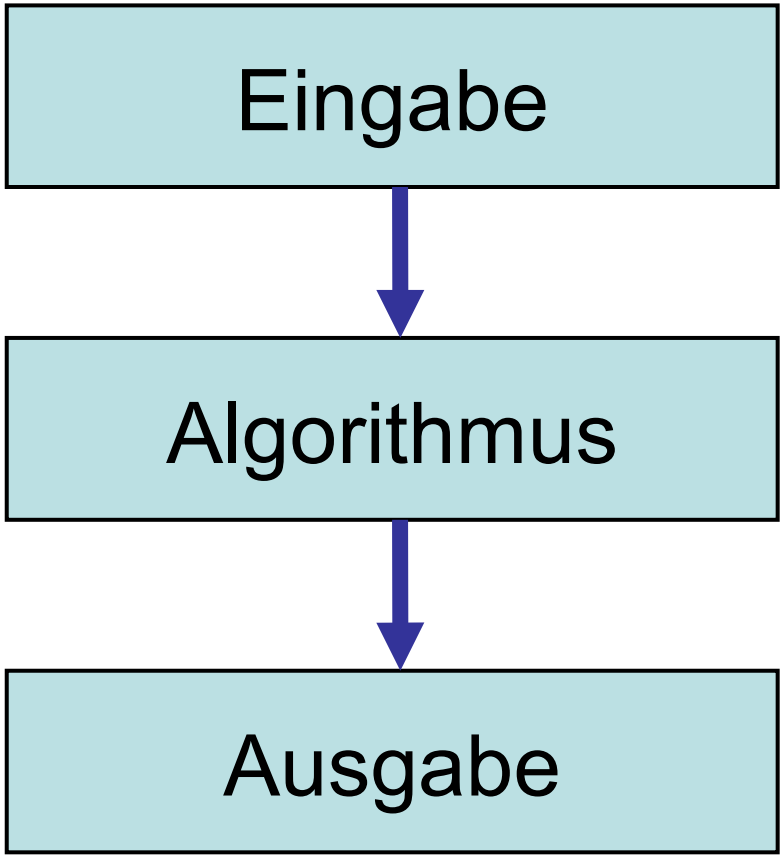
Was ist ein Algorithmus?

Definition: Ein Algorithmus ist eine formale Handlungsvorschrift zur Lösung von Instanzen eines Problems in endlich vielen Schritten.

Algorithmus: von Muhammad Al Chwarizmi (~783-850), dessen arabisches Lehrbuch "Über das Rechnen mit indischen Zahlen" in lateinischer Übersetzung mit "Dixit Algorismi" begann. Im Mittelalter wurde daraus "algorismus" als Bezeichnung für die Kunst des Rechnens mit den arabischen (bzw. indischen) Ziffern

"algorismus" $\longrightarrow$ "Algorithmus"

Was ist ein Algorithmus?



Beispiele

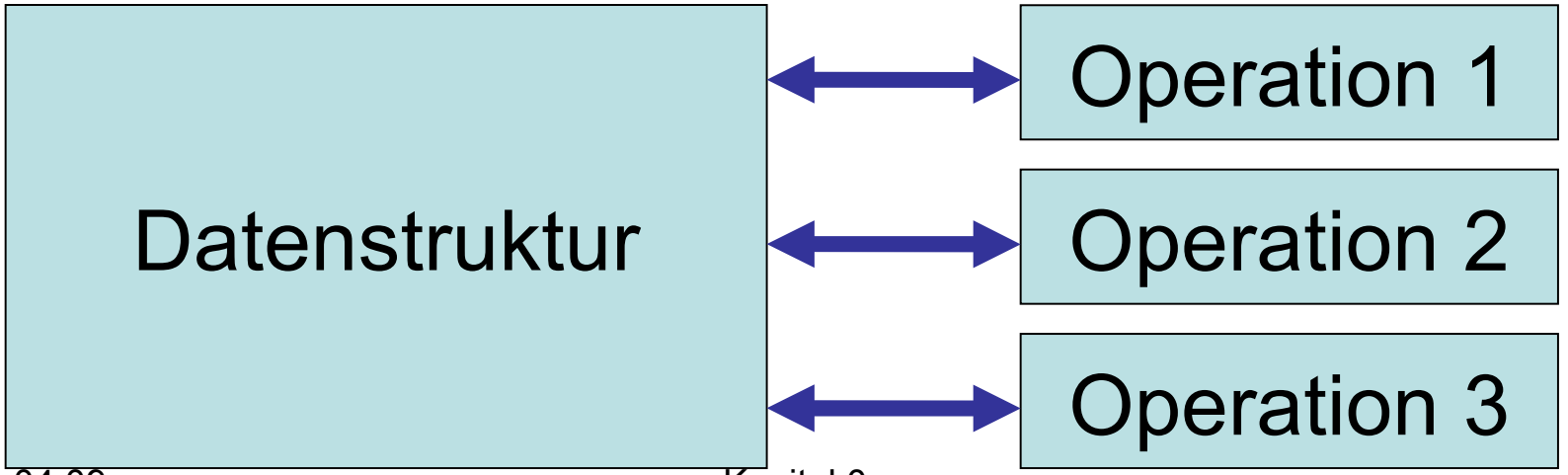
- Kochrezept
  Eingabe: Zutaten, Algorithmus: Rezept,
  Ausgabe: Essen
- Bauanleitung
  Eingabe: Teile, Algorithmus: Bauanleitung,
  Ausgabe: Schrank

Weitere Beispiele (später):

- Ausweg aus Labyrinth
- Zeichnen eines Kreises

Was ist eine Datenstruktur?

Definition: Eine Datenstruktur ist eine Anordnung / Verknüpfung von Daten, um den Zugriff auf diese und ihre Verwaltung geeignet zu ermöglichen.

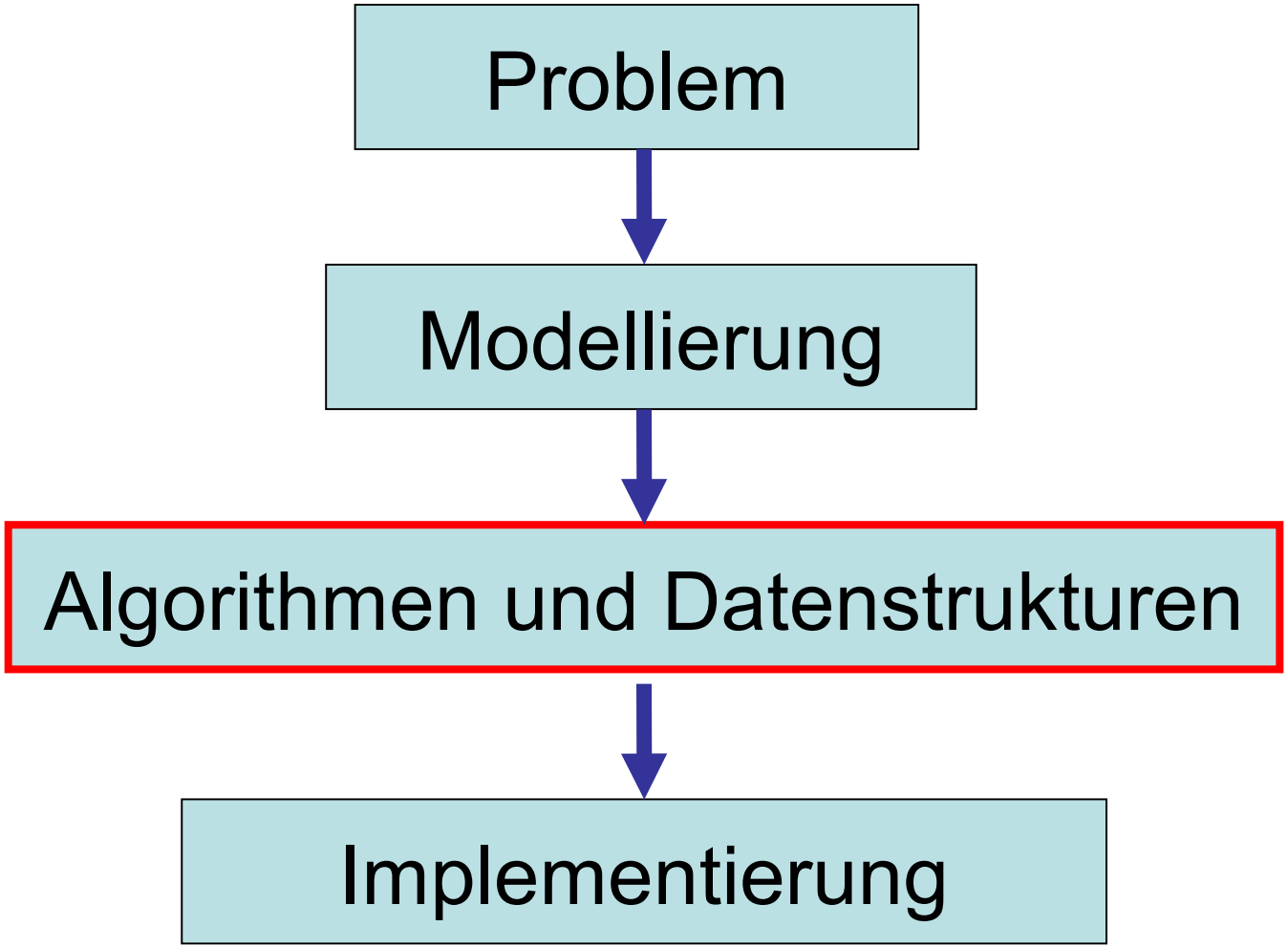


Beispiele

Datenstrukturen für schnelles Suchen:

- Lexikon
  Operation: Suche(Name)
  (Algo mit Eingabe “Name”, Ausgabe Info zu “Name”)
- Kalender
  Operation: z.B. Wochentag(Datum)
  (Algo mit Eingabe “Datum”, Ausgabe Wochentag des Datums)

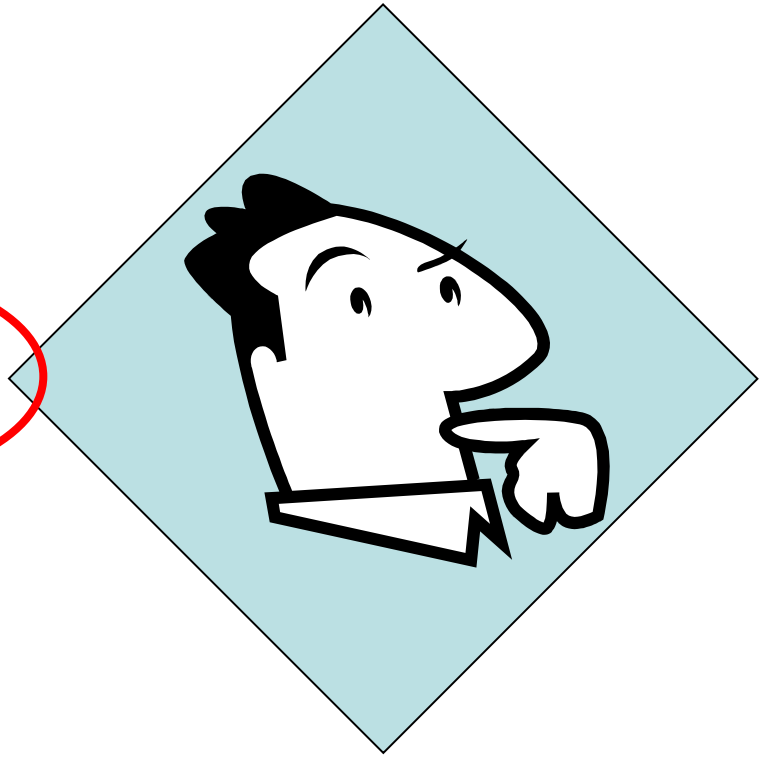
Softwareentwicklung



Grundsätzliche Probleme

Korrektheit

Effizienz



Komplexität

Robustheit / Sicherheit

Effizienz

Wichtig: Laufzeit und Speicheraufwand

Warum?

- Riesige Datenmengen (Bioinformatik)
- Echtzeitanwendungen (Spiele)

Ziel der Vorlesung:

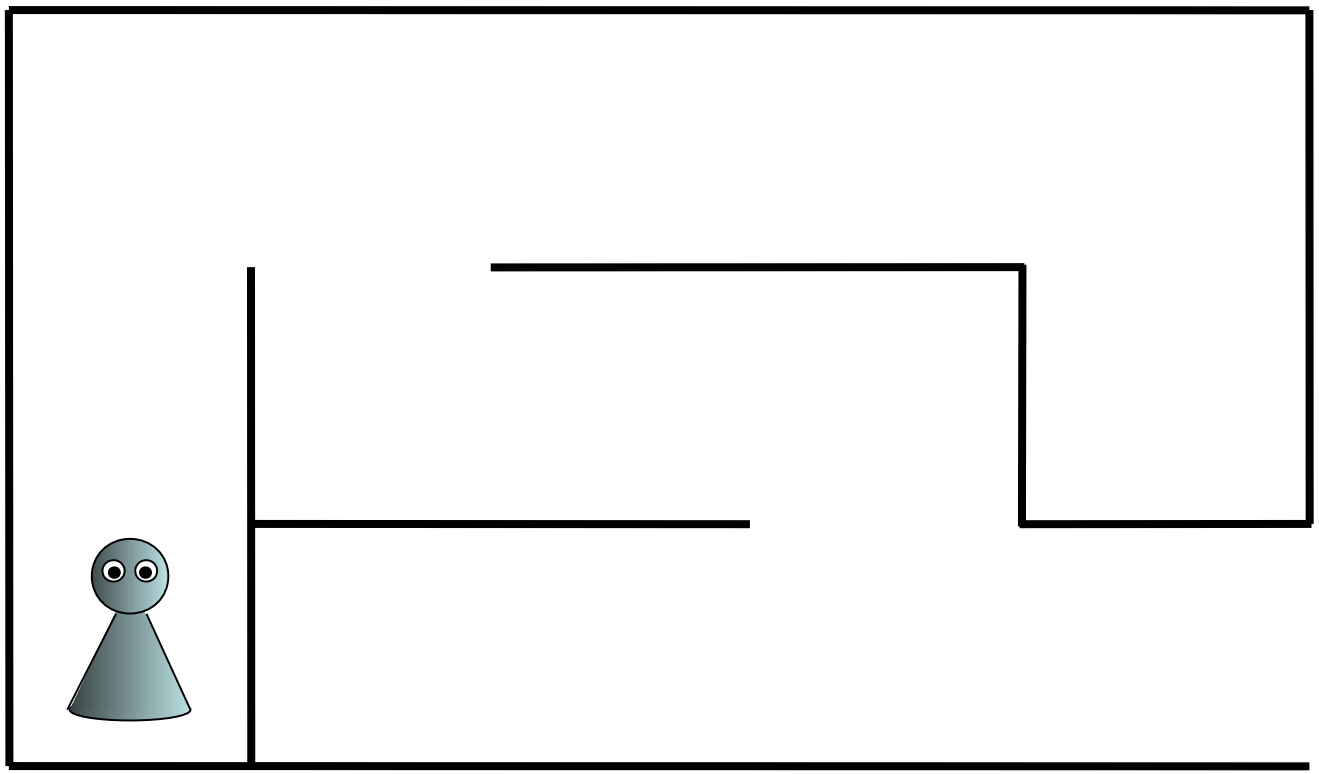
Grundstock an effizienten Algorithmen und Datenstrukturen für Standardprobleme

Beispiele

Einige sehr effiziente Algorithmen:

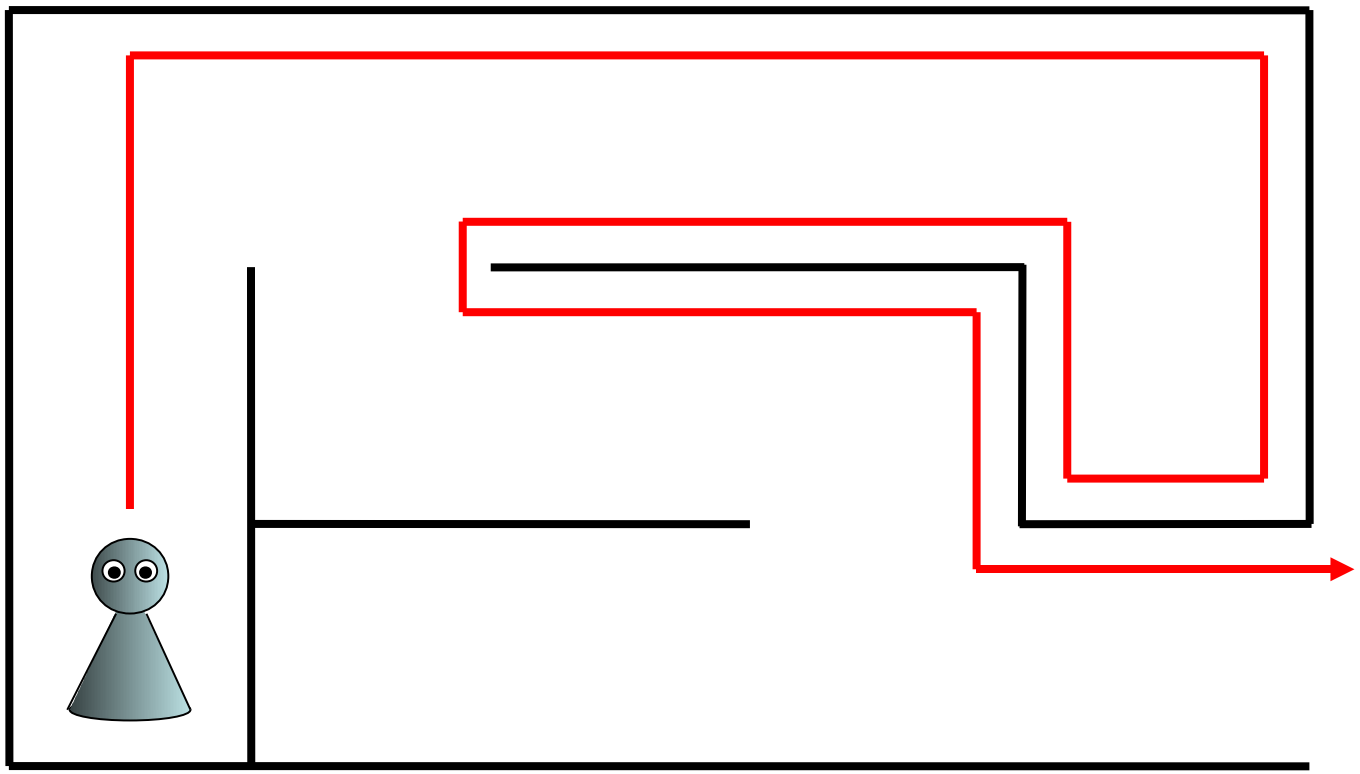
- Ausweg aus Labyrinth
- Zeichnen eines Kreises

Ausweg aus Labyrinth



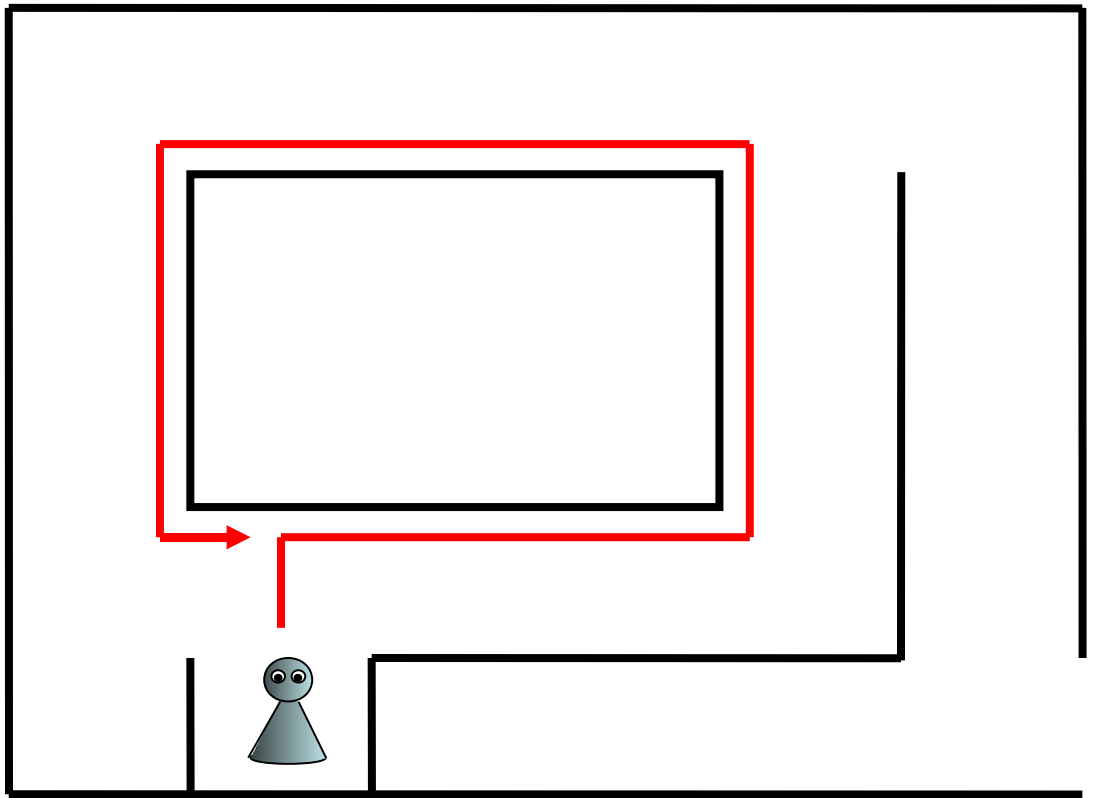
Es ist dunkel! Keine Taschenlampe!

Ausweg aus Labyrinth



1. Versuch: Linke-Hand-Regel

Ausweg aus Labyrinth



Zwecklos! Ich bin eingemauert…

Ausweg aus Labyrinth

Pledge-Algorithmus:

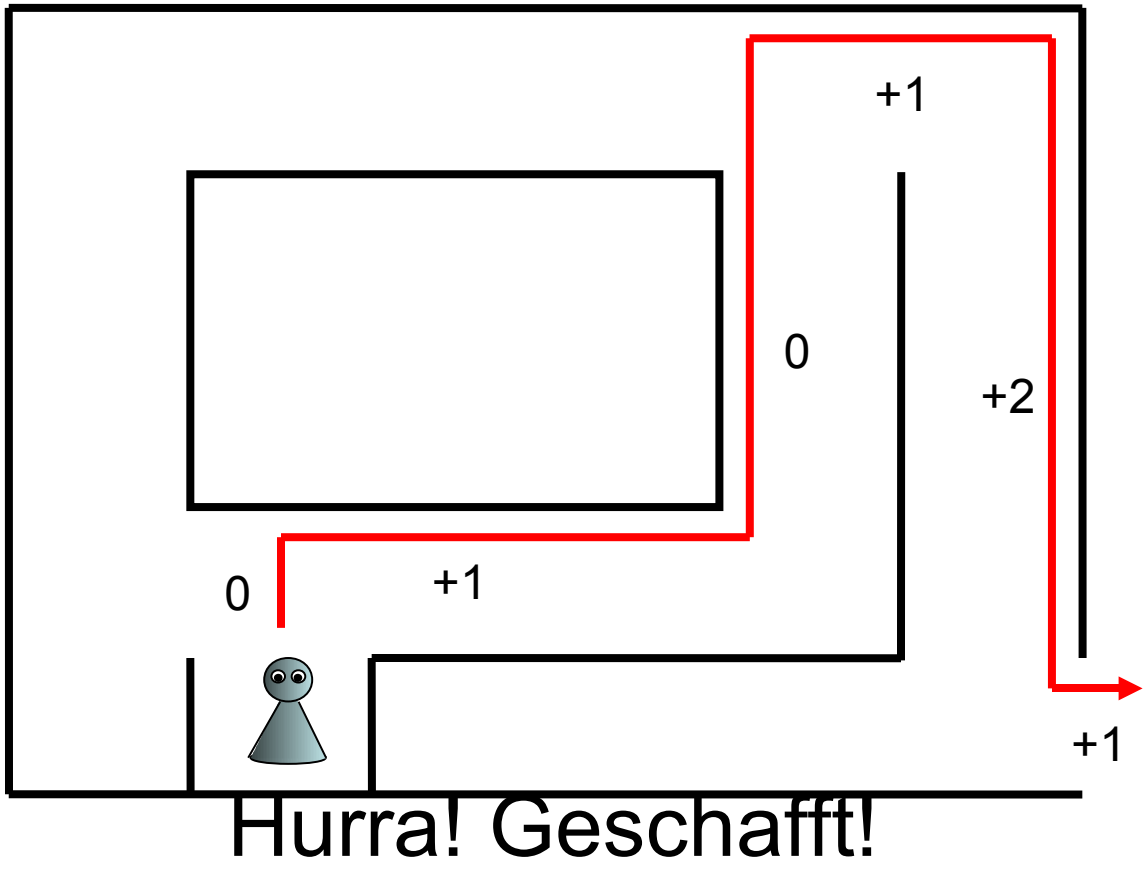
setze Umdrehungszähler auf 0

wiederhole, bis Ausgang gefunden

- geradeaus, bis Wand erreicht
- Linke-Hand-Regel bis Umdrehungszähler=0
  (Linksdrehung um 90 Grad: Zähler-1,
  Rechtsdrehung um 90 Grad: Zähler+1)

Funktioniert, solange das Labyrinth rechtwinklig und eben und der Ausgang an der Seite ist!

Ausweg aus Labyrinth



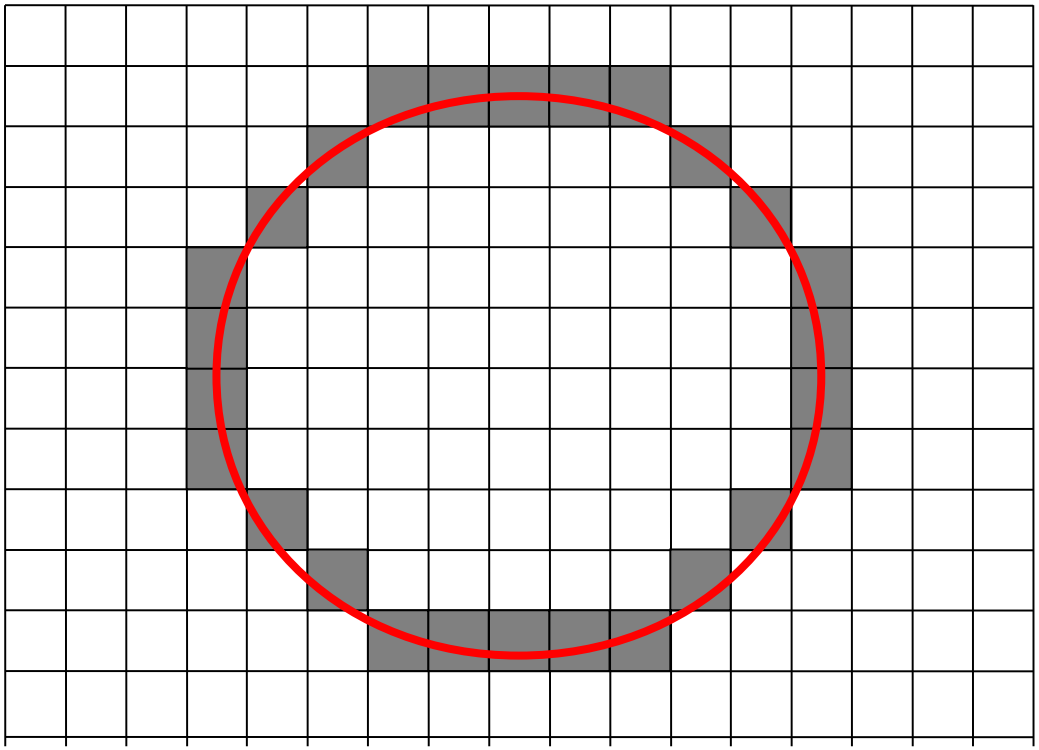
Algorithmusbuch

Einige dieser Algorithmen sind erschienen in

B. Vöcking, H. Alt, M. Dietzfelbinger C. Scheideler, H. Vollmer, D. Wagner
Taschenbuch der Algorithmen
Springer Verlag, Berlin, 2008

Kreiszeichnen

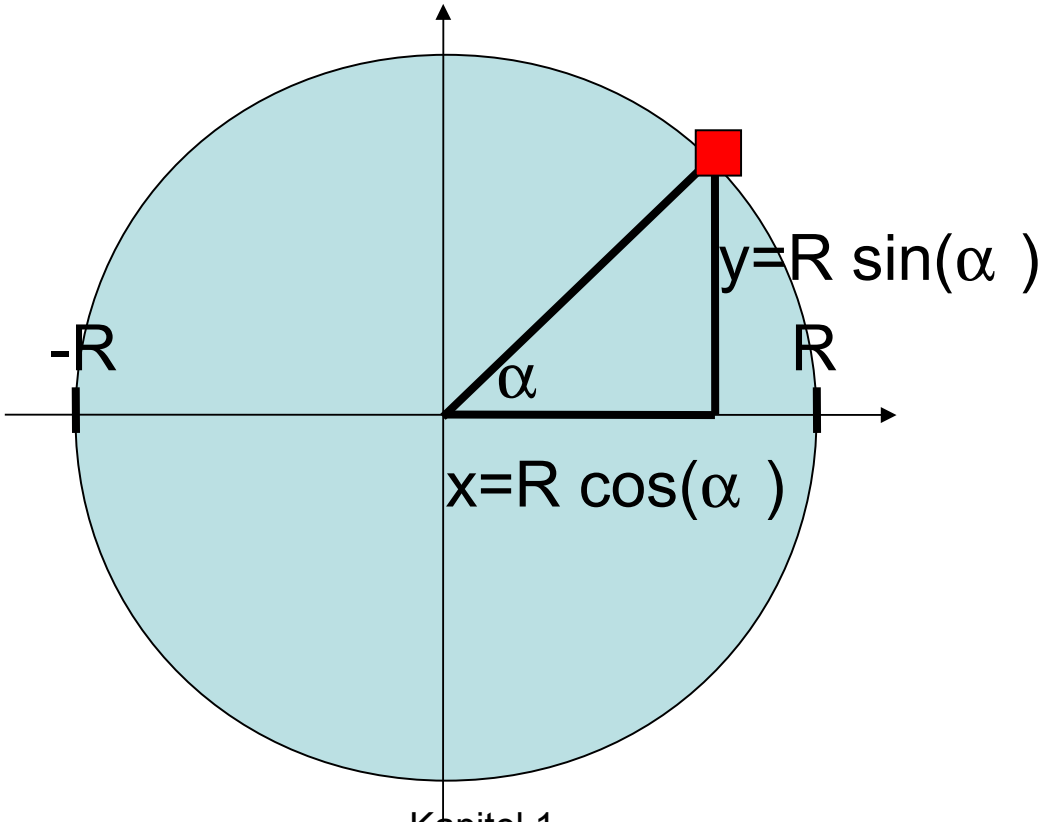
Wie zeichne ich schnell Kreise im Computer?



Kreiszeichnen

Naiver Ansatz: verwende sin und cos

Für $\alpha = 0...2\pi$:



Kreiszeichnen

plot(x,y): setze Punkt bei Position (x,y)

Algorithmus Kreis1:

```
for (1=0; i<n; i++)
plot(R*cos(2*pi*i/n), R*sin(2*pi*i/n));
```

Kreisumfang: $U=2\pi R$. Bei Pixelbreite von 1 Längeneinheit reicht dann $n=7R$.

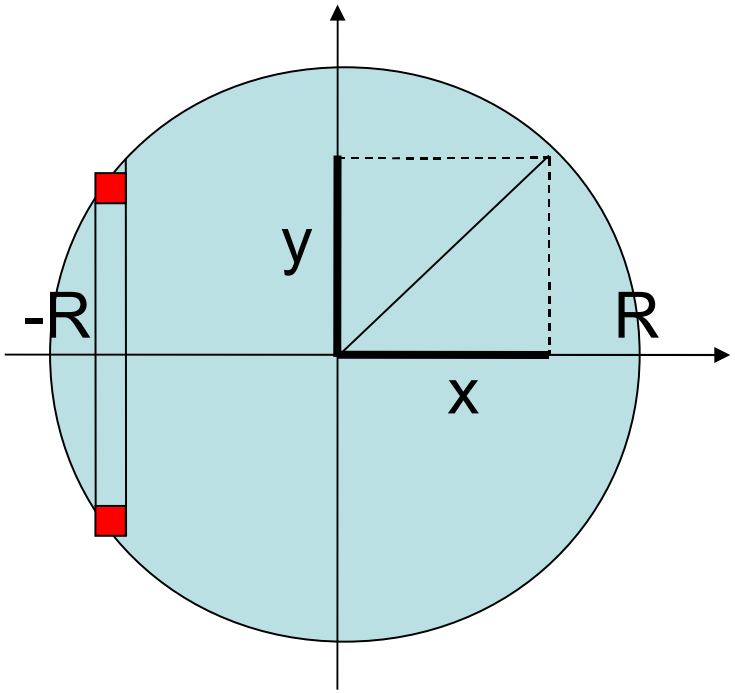
Problem: Funktionen sin, cos sind teuer!

Kreiszeichnen

Besserer Ansatz:
$x^2+y^2=R^2$, also $y=\pm\mathtt{sqrt}(R^2-x^2)$

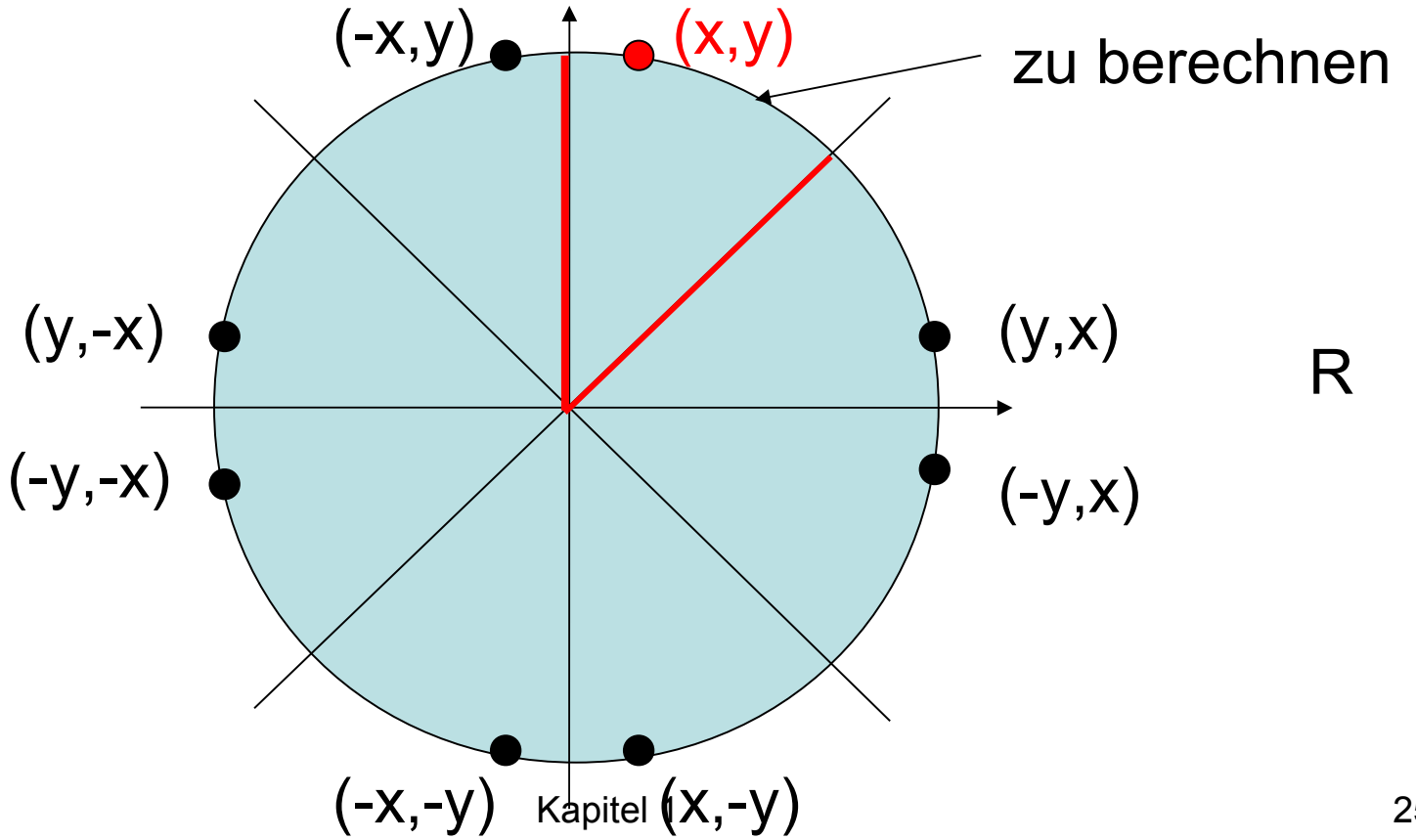
Algorithmus Kreis2:

```
for(x=-R;x<=R;x++){
y=sqrt(R*R-x*x);
   plot(x,y);
   plot(x,-y);
  }
```



Kreiszeichnen

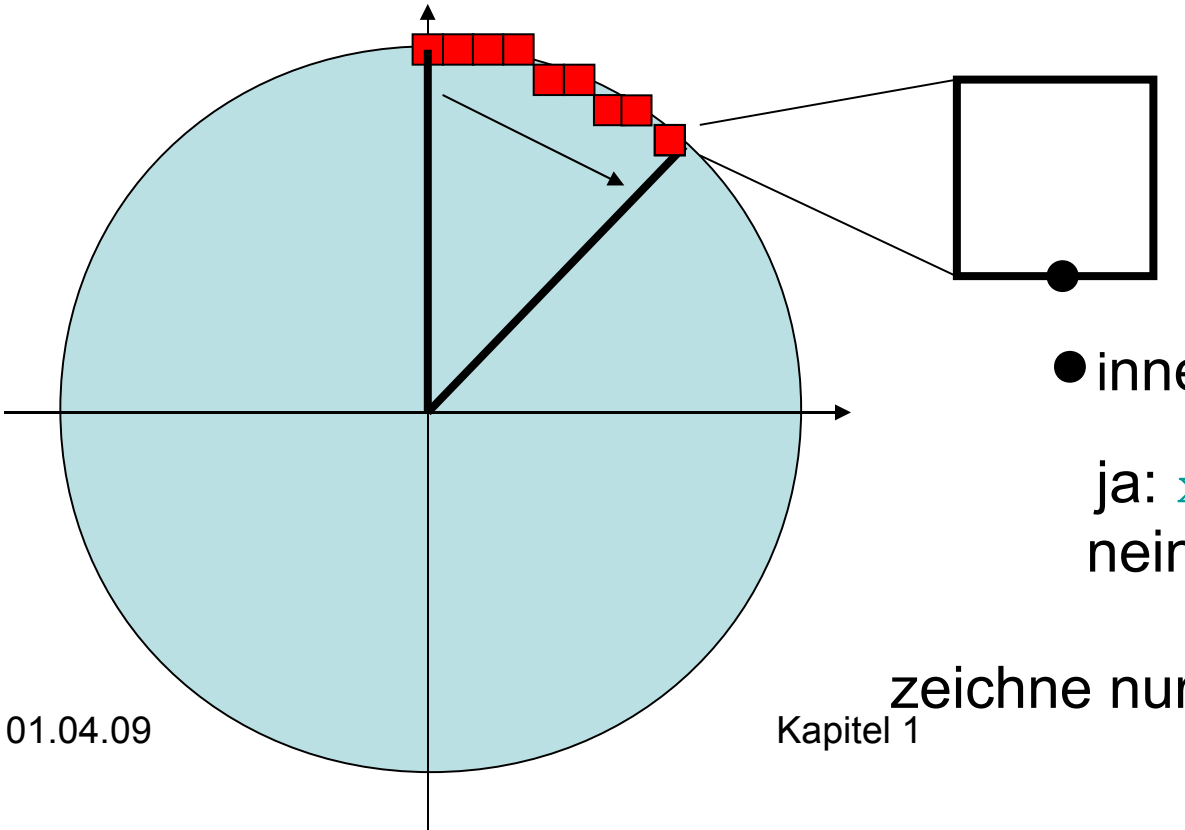
Noch besser: Ausnutzung von Spiegelungen



Kreiszeichnen

Aber: Algo verwendet immer noch Wurzel

Kommen wir mit *,+,- aus?



● innerhalb von Kreis?

ja: `x=x+1;`

nein: `x=x+1; y=y-1;`

zeichne nur ■ mit ● innerhalb

Kreiszeichnen

- Mittelpunkt des ersten ■ $(x,y)=(0,R)$
- Position seines Punkts: $(0,R-1/2)$ (1x1-Quadrat)
- Test, ob (x,y) innerhalb Kreis:
  teste, ob $F(x,y):= x^2+y^2-R^2 < 0$ ist
- Also Test, ob ■ mit Punkt $(0,R-1/2)$ noch innerhalb des Kreises ist: $0^2+(R-1/2)^2-R^2 = -R+0.25 < 0$?
- Es gilt: $F(x+1,y) = F(x,y)+2x+1$
  $F(x+1,y-1) = F(x,y)+2x-2y+2$

Bresenham Algorithmus

```
x=0; y=R; F=-R+0.25;
plot(0,R); plot(R,0); plot(0,-R); plot(-R,0);

while (x<y) {
   x=x+1;
   F=F+2*x-1;
if (F>=0) {  // ■ bei (x,y) hat ● nicht innerhalb
     y=y-1;   //  deshalb zu (x,y-1)
     F=F-2*y;
   }
plot(x,y); plot(y,x); plot(-x,y); plot(y,-x);
plot(x,-y); plot(-y,x); plot(-x,-y); plot(-y,-x);
}
```

Nächstes Kapitel

- Eingabekodierung
- Asymptotische Notation
- Maschinenmodell
- Java
- Laufzeitanalyse
- Einige Beispiele