

# Algorithmen

Lukas Abelt

`lukas.abelt@airbus.com`

DHBW Ravensburg  
Wirtschaftsinformatik

Ravensburg  
27. März 2019

# Outline

- 1 Allgemeines
  - Begriffsklärung
  - Ziele des Moduls
- 2 Beschreibung
  - Formale Eigenschaften
  - Darstellungsformen
- 3 Analyse
  - Korrektheit eines Algorithmus
  - Komplexitätsanalyse

# Inhalt

## 1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

## 2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

## 3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

# Inhalt

## 1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

## 2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

## 3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

# Begriffklärung

## Etymologie

- Leitet sich ursprünglich vom persischen Astronomen „Muhammad Ibn-Musa al-Hwarizmi“ ab

# Begriffklärung

## Etymologie

- Leitet sich ursprünglich vom persischen Astronomen „Muhammad Ibn-Musa al-Hwarizmi“ ab
  - Schrieb Bücher über das indische Zahlensystem (um 800 n. Chr.)

# Begriffklärung

## Etymologie

- Leitet sich ursprünglich vom persischen Astronomen „Muhammad Ibn-Musa al-Hwarizmi“ ab
  - Schrieb Bücher über das indische Zahlensystem (um 800 n. Chr.)
  - Im 12. Jh übersetzt ins lateinische

# Begriffklärung

## Etymologie

- Leitet sich ursprünglich vom persischen Astronomen „Muhammad Ibn-Musa al-Hwarizmi“ ab
  - Schrieb Bücher über das indische Zahlensystem (um 800 n. Chr.)
  - Im 12. Jh übersetzt ins lateinische
  - Dabei wurde der Namensbestandteil „al-Hwarizmi“ in „Algorismi“ lateinisiert



# Begriffklärung

## Etymologie

- Leitet sich ursprünglich vom persischen Astronomen „Muhammad Ibn-Musa al-Hwarizmi“ ab
  - Schrieb Bücher über das indische Zahlensystem (um 800 n. Chr.)
  - Im 12. Jh übersetzt ins lateinische
  - Dabei wurde der Namensbestandteil „al-Hwarizmi“ in „Algorismi“ lateinisiert
- Durch spätere Überlieferungen wurde der Begriff später als Zusammensetzung betrachtet aus...

# Begriffklärung

## Etymologie

- Leitet sich ursprünglich vom persischen Astronomen „Muhammad Ibn-Musa al-Hwarizmi“ ab
  - Schrieb Bücher über das indische Zahlensystem (um 800 n. Chr.)
  - Im 12. Jh übersetzt ins lateinische
  - Dabei wurde der Namensbestandteil „al-Hwarizmi“ in „Algorismi“ lateinisiert
- Durch spätere Überlieferungen wurde der Begriff später als Zusammensetzung betrachtet aus...
  - Dem Namen „Algus-“...

# Begriffklärung

## Etymologie

- Leitet sich ursprünglich vom persischen Astronomen „Muhammad Ibn-Musa al-Hwarizmi“ ab
  - Schrieb Bücher über das indische Zahlensystem (um 800 n. Chr.)
  - Im 12. Jh übersetzt ins lateinische
  - Dabei wurde der Namensbestandteil „al-Hwarizmi“ in „Algorismi“ lateinisiert
- Durch spätere Überlieferungen wurde der Begriff später als Zusammensetzung betrachtet aus...
  - Dem Namen „Algus-“...
  - und dem aus dem griechisch entlehnten „-rismus“ (Zahl)

# Begriffsklärung

Was bedeutet das jetzt

## Formale Definition

*Eine Berechnungsvorschrift zur Lösung eines Problems heißt genau dann Algorithmus, wenn eine zu dieser Berechnungsvorschrift äquivalente Turingmaschine existiert, die für jede Eingabe, die eine Lösung besitzt, stoppt.*

# Begriffsklärung

Was bedeutet das jetzt

## Formale Definition

*Eine Berechnungsvorschrift zur Lösung eines Problems heißt genau dann Algorithmus, wenn eine zu dieser Berechnungsvorschrift äquivalente Turingmaschine existiert, die für jede Eingabe, die eine Lösung besitzt, stoppt.*

## Oder auch

*Ein Algorithmus ist eine domänenunabhängige Beschreibung einer Handlungsvorschrift zur Lösung eines Problems. Eine bestimmte Eingabe wird in eine bestimmte Ausgabe überführt.*

# Begriffsklärung

Also

- Ist also die Beschreibung eines Programmes oder einer Funktion

# Begriffsklärung

## Also

- Ist also die Beschreibung eines Programmes oder einer Funktion
  - Unabhängig von der verwendeten Programmiersprache!

# Begriffsklärung

## Also

- Ist also die Beschreibung eines Programmes oder einer Funktion
  - Unabhängig von der verwendeten Programmiersprache!
  - Source Code direkt ist also kein Algorithmus...



# Begriffsklärung

## Also

- ▣ Ist also die Beschreibung eines Programmes oder einer Funktion
  - ▣ Unabhängig von der verwendeten Programmiersprache!
  - ▣ Source Code direkt ist also kein Algorithmus...
  - ▣ ...aber aus diesem lässt sich der verwendete Algorithmus ableiten und beschreiben

# Begriffsklärung

## Also

- ▣ Ist also die Beschreibung eines Programmes oder einer Funktion
  - ▣ Unabhängig von der verwendeten Programmiersprache!
  - ▣ Source Code direkt ist also kein Algorithmus...
  - ▣ ...aber aus diesem lässt sich der verwendete Algorithmus ableiten und beschreiben
- ▣ Algorithmen können in verschiedenen Formen dargestellt werden (Mehr dazu im nächsten Kapitel)

# Inhalt

## 1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

## 2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

## 3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

# Ziele

- Am Ende des Moduls könnt ihr...

# Ziele

- Am Ende des Moduls könnt ihr...
  - Einen Algorithmus in eine Implementierung umsetzen

# Ziele

- ▣ Am Ende des Moduls könnt ihr...
  - ▣ Einen Algorithmus in eine Implementierung umsetzen
  - ▣ Aus einer Implementierung den Algorithmus ableiten

# Ziele

- Am Ende des Moduls könnt ihr...
  - Einen Algorithmus in eine Implementierung umsetzen
  - Aus einer Implementierung den Algorithmus ableiten
  - Die formalen Eigenschaften von Algorithmen kennen

# Ziele

- Am Ende des Moduls könnt ihr...
  - Einen Algorithmus in eine Implementierung umsetzen
  - Aus einer Implementierung den Algorithmus ableiten
  - Die formalen Eigenschaften von Algorithmen kennen
  - Algorithmen anhand der kennengelernten Methoden zu analysieren



# Inhalt

## 1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

## 2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

## 3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

# Inhalt

## 1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

## 2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

## 3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

# Eigenschaften von Algorithmen

## Grundlegendes

- ▣ **Finitheit** - Ein Algorithmus lässt sich in endlich vielen Schritten eindeutig beschreiben

# Eigenschaften von Algorithmen

## Grundlegendes

- **Finitheit** - Ein Algorithmus lässt sich in endlich vielen Schritten eindeutig beschreiben
- **Ausführbarkeit** - Jeder Einzelschritt muss tatsächlich ausführbar sein

# Eigenschaften von Algorithmen

## Grundlegendes

- **Finitheit** - Ein Algorithmus lässt sich in endlich vielen Schritten eindeutig beschreiben
- **Ausführbarkeit** - Jeder Einzelschritt muss tatsächlich ausführbar sein
- **Platzkomplexität** - Ein Algorithmus benötigt zu jedem Zeitpunkt nur endlich viel Speicherplatz

# Eigenschaften von Algorithmen

## Grundlegendes

- **Finitheit** - Ein Algorithmus lässt sich in endlich vielen Schritten eindeutig beschreiben
- **Ausführbarkeit** - Jeder Einzelschritt muss tatsächlich ausführbar sein
- **Platzkomplexität** - Ein Algorithmus benötigt zu jedem Zeitpunkt nur endlich viel Speicherplatz
- **Terminierung** - Der Algorithmus benötigt eine endliche Anzahl von Schritten zur Ausführung

# Eigenschaften von Algorithmen

## Grundlegendes

- **Finitheit** - Ein Algorithmus lässt sich in endlich vielen Schritten eindeutig beschreiben
- **Ausführbarkeit** - Jeder Einzelschritt muss tatsächlich ausführbar sein
- **Platzkomplexität** - Ein Algorithmus benötigt zu jedem Zeitpunkt nur endlich viel Speicherplatz
- **Terminierung** - Der Algorithmus benötigt eine endliche Anzahl von Schritten zur Ausführung
- **Determiniertheit** - Der Algorithmus muss bei gleichen Rahmenbedingungen das gleiche Ergebnis liefern

# Eigenschaften von Algorithmen

## Grundlegendes

- **Finitheit** - Ein Algorithmus lässt sich in endlich vielen Schritten eindeutig beschreiben
- **Ausführbarkeit** - Jeder Einzelschritt muss tatsächlich ausführbar sein
- **Platzkomplexität** - Ein Algorithmus benötigt zu jedem Zeitpunkt nur endlich viel Speicherplatz
- **Terminierung** - Der Algorithmus benötigt eine endliche Anzahl von Schritten zur Ausführung
- **Determiniertheit** - Der Algorithmus muss bei gleichen Rahmenbedingungen das gleiche Ergebnis liefern
- **Determinismus** - Der nächste Schritt des Algorithmus ist zu jedem Zeitpunkt genau definiert



# Korrektheit von Algorithmen

- Jeder Algorithmus sollte auch in allen Fällen das korrekte Ergebnis liefern...

# Korrektheit von Algorithmen

- Jeder Algorithmus sollte auch in allen Fällen das korrekte Ergebnis liefern...
- Klingt simpel, aber eindeutiger Beweis für alle Eingaben oft schwierig

# Korrektheit von Algorithmen

- Jeder Algorithmus sollte auch in allen Fällen das korrekte Ergebnis liefern...
- Klingt simpel, aber eindeutiger Beweis für alle Eingaben oft schwierig
- Testen an ausgewählten Beispielen **nicht** ausreichend

# Korrektheit von Algorithmen

- Jeder Algorithmus sollte auch in allen Fällen das korrekte Ergebnis liefern...
- Klingt simpel, aber eindeutiger Beweis für alle Eingaben oft schwierig
- Testen an ausgewählten Beispielen **nicht** ausreichend
  - Jedoch verringern umfangreiche Tests natürlich das Risiko eines unentdeckten Fehler

# Korrektheit von Algorithmen

- Jeder Algorithmus sollte auch in allen Fällen das korrekte Ergebnis liefern...
- Klingt simpel, aber eindeutiger Beweis für alle Eingaben oft schwierig
- Testen an ausgewählten Beispielen **nicht** ausreichend
  - Jedoch verringern umfangreiche Tests natürlich das Risiko eines unentdeckten Fehler
- Korrektheit lässt sich im Grunde nur durch formalen Beweis zeigen

# Korrektheit von Algorithmen

- Jeder Algorithmus sollte auch in allen Fällen das korrekte Ergebnis liefern...
- Klingt simpel, aber eindeutiger Beweis für alle Eingaben oft schwierig
- Testen an ausgewählten Beispielen **nicht** ausreichend
  - Jedoch verringern umfangreiche Tests natürlich das Risiko eines unentdeckten Fehler
- Korrektheit lässt sich im Grunde nur durch formalen Beweis zeigen
  - Diese sind häufig sehr umfangreich und komplex...

# Korrektheit von Algorithmen

- Jeder Algorithmus sollte auch in allen Fällen das korrekte Ergebnis liefern...
- Klingt simpel, aber eindeutiger Beweis für alle Eingaben oft schwierig
- Testen an ausgewählten Beispielen **nicht** ausreichend
  - Jedoch verringern umfangreiche Tests natürlich das Risiko eines unentdeckten Fehler
- Korrektheit lässt sich im Grunde nur durch formalen Beweis zeigen
  - Diese sind häufig sehr umfangreich und komplex...
  - ...und deshalb auch nicht Teil der Vorlesung

# Korrektheit von Algorithmen



Quelle:

*„Program testing can be used to show the presence of bugs, but never to show their absence!“*

Edsger W. Dijkstra



# Effizienz von Algorithmen

- Ergibt sich indirekt aus den Grundlegenden Eigenschaften
- Effizienz lässt sich über verschiedene Größen beschreiben:
  - Speicherverbrauch
  - Zeitverbrauch
- Die sind jedoch oft Implementierungs- und Rechnerabhängig
- Deshalb wird mit formalisierten Modellen gearbeitet
- ...Mehr dazu im Kapitel „Analyse“

# Inhalt

## 1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

## 2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

## 3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse



# Inhalt

## 1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

## 2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

## 3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

# Inhalt

## 1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

## 2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

## 3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

# Inhalt

## 1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

## 2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

## 3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

# Kontakt

- E-Mail: `lukas.abelt@airbus.com`
- GitHub: `https://www.github.com/LuAbelt`
- GitLab: `https://www.gitlab.com/LuAbelt`
- Telefon(Firma): 07545 - 8 8895
- Telegram: LuAbelt