

# Detaillierte Gliederung

## Algorithmen und Datenstrukturen (für ET/IT)

Sommersemester 2014

Dr. Tobias Lasser

Stand: 10. Juli 2014

### 1 Einführung

- Wo kommen Algorithmen und Datenstrukturen vor?
- Definition und Eigenschaften Algorithmus, Beispiele
- Definition Datenstruktur und Beispiel
- Ebenen eines Computers, Einordnung Algorithmen und Datenstrukturen

### 2 Grundlagen von Algorithmen

#### 2.1 Darstellung von Algorithmen

- Darstellung von Algorithmen mittels Pseudocode, Flussdiagrammen, Struktogrammen und Programmiersprachen
- Übersicht von Programmiersprachen
- Äquivalenz von Algorithmen-Beschreibungen (Churchsche These)
- Beispiel Euklidischer Algorithmus

#### 2.2 Elementare Bausteine

- Die vier elementaren Bausteine von Algorithmen (elementarer Verarbeitungsschritt, Sequenz, bedingter Verarbeitungsschritt, Wiederholung)
- Repräsentation der vier elementaren Bausteine als Pseudocode
- Pseudocode-Konventionen in der Vorlesung
- Beispiel rekursive Berechnung der Fibonacci-Zahlen

#### 2.3 Logische Ausdrücke

- Logische Werte und Verknüpfungen (AND, OR, NOT), Wahrheitstabellen
- Weitere Verknüpfungen: NAND, NOR, XOR, Implikation, Äquivalenz
- Rangfolge und Rechenregeln
- Logische Ausdrücke in Pseudocode und C++

## **3 Grundlagen von Datenstrukturen**

### **3.1 Primitive Datentypen und Zahldarstellung**

- Primitive Datentypen
- Bits und Bytes, Größenangaben
- Dezimalsystem, Binärsystem
- Oktalsystem, Hexadezimalsystem
- Anzahl Ziffern pro Zahl
- Größte Zahl pro Anzahl Ziffern
- Negative Zahlen: 2-Komplement Darstellung
- Rationale Zahlen: Festkomma Darstellung
- Floating Point Zahlen
- Primitive Datentypen mit Operationen

### **3.2 Felder als sequentielle Liste**

- Definition Feld
- Feld als sequentielle Liste, Eigenschaften
- Operationen auf sequentiellen Listen (verlängern, löschen, einfügen)
- Optional: Beispiele für sequentielle Listen in C++

### **3.3 Zeichen und Zeichenfolgen**

- ASCII Code, Unicode
- Zeichen und Zeichenfolgen (Strings)
- Optional: Strings in C++

### **3.4 Felder als verkettete Liste**

- Feld als verkettete Liste
- Operationen auf verketteten Listen (Zugriff, löschen, einfügen)
- Optional: Verkettete Listen in C++
- Feld als doppelt verkettete Liste mit Operationen und Eigenschaften
- Optional: Doppelt verkettete Listen in C++

### **3.5 Abstrakte Datentypen**

- Definition Abstrakter Datentyp
- Abstrakte Variable
- Abstrakte Liste

### **3.6 Stack**

- Definition Stack mit Operationen, Stack als abstrakter Datentyp
- Anwendungsbeispiele Stack
- Implementationen von Stack als sequentielle Liste, verkettete Liste

### **3.7 Queue**

- Definition Queue mit Operationen, Queue als abstrakter Datentyp
- Anwendungsbeispiele Queue
- Implementationen von Queue als verkettete Liste, (zirkuläre) sequentielle Liste, mit zwei Stacks

## **4 Grundlagen der Korrektheit von Algorithmen**

### **4.1 Motivation und Spezifikation**

- Beispiele von bekannten Software-Fehlern
- Relative Korrektheit, Nachweis von Korrektheit durch Verifikation, Validation

### **4.2 Verifikation**

- Vor- und Nachbedingungen, partielle und totale Korrektheit
- Korrektheit von Anweisungstypen (4 elementare Bausteine)
- Schleifeninvarianten, Nachweis Korrektheit und Beispiele

### **4.3 Beispiel: Insertion Sort**

- Algorithmus Insertion Sort in Pseudocode
- Beispielablauf Insertion Sort
- Verifikation von Insertion Sort mit Invariante

### **4.4 Validation**

- Validation durch systematische Tests: Blackbox-, Whitebox-, Regression-, Integrations-Test
- Fehlerquellen, fehlertolerantes Programmieren, fehlerpräventives Programmieren

## **5 Grundlagen der Effizienz von Algorithmen**

### **5.1 Motivation**

- Komplexität von Insertion Sort
- Wachstumsraten mit Illustrationen

### **5.2 RAM-Modell**

- Rechnermodell zur Laufzeitanalyse
- Annahmen des RAM-Modells

### 5.3 Landau-Symbole

- Landau-Symbol  $\Theta$ , Definition und Beispiele
- Landau-Symbol  $O$ , Definition und Beispiele
- Kategorisierung der Effizienz von Algorithmen mittels  $O$
- Konstanten in Landau-Symbolen
- Rechenregel für Addition in  $O$ -Notation
- Komplexität der elementaren Bausteine sowie von Algorithmen
- Komplexität der behandelten elementaren Datenstrukturen mit Operationen

## 6 Grundlagen des Algorithmen-Entwurfs

### 6.1 Entwurfsprinzipien

- Entwurfsprinzip Verfeinerung
- Entwurfsprinzip Algorithmen-Muster

### 6.2 Divide and Conquer

- Definition und Pseudocode Divide and Conquer
- Anwendungsbeispiel: MergeSort
- Komplexität MergeSort mit Rekursionsbaum
- Anwendungsbeispiel: QuickSort mit Eigenschaften und Komplexität
- Übersicht behandelte Sortierverfahren

### 6.3 Greedy-Algorithmen

- Algorithmen-Muster Greedy, Pseudocode
- Anwendungsbeispiel: Wechselgeld
- Anwendungsbeispiel: Glasfasernetz, minimaler Spannbaum

### 6.4 Backtracking

- Algorithmen-Muster Backtracking, Pseudocode
- Anwendungsbeispiel: Labyrinth, Maus mit Käse
- Anwendungsbeispiel: Traveling Salesman Problem
- Anwendungsbeispiel: Acht-Damen-Problem

### 6.5 Dynamisches Programmieren

- Prinzip dynamisches Programmieren
- Fibonacci-Folge Definition, rekursive Berechnung
- Anwendungsbeispiel: Fibonacci-Folge mit dynamischer Programmierung

## 7 Fortgeschrittene Datenstrukturen

### 7.1 Graphen

- Definition Graph (gerichtet, ungerichtet, gewichtet)
- Graph-Eigenschaften wie Eingangs-/Ausgangsgrad, Pfad und Länge, Zyklus, Zusammenhang
- Darstellung mit Adjazenzmatrix und Adjazenzliste, Komplexität einfacher Operationen

### 7.2 Bäume

- Definition Baum und Wald, Eigenschaften
- Wurzel von Bäumen, Tiefe und Ebenen, Höhe
- Spezialfall Binärbaum
- Darstellung von Bäumen mit angepassten verketteten Listen bzw. doppelt verketteten Listen
- Binärbäume als sequentielle Listen
- Traversierung von Bäumen mit DFS (Pre-order, In-order, Post-order), BFS
- Anwendung: Quadtree

### 7.3 Heaps

- Definition Heap, Eigenschaften
- Operationen extractMin, minHeapify, buildMinHeap
- HeapSort mit Eigenschaften und Komplexität
- Stabilität von Sortierverfahren, Übersicht über behandelte Sortierverfahren

### 7.4 Priority Queues

- Definition Priority Queue mit Operationen, Priority Queue als abstrakter Datentyp
- Implementation als sortiertes oder unsortiertes Feld und Komplexitäten
- Implementation als Heap und Komplexitäten, Operationen insert und decreaseKey
- Sortieren mit Priority Queues

## 8 Such-Algorithmen

### 8.1 Lineare Suche

- Lineare Suche Algorithmus mit Komplexität

### 8.2 Binäre Suche

- Binäre Suche Algorithmus mit Komplexität
- Implementierung als rekursiver / iterativer Algorithmus

### 8.3 Suchbäume

- Definition binärer Suchbaum mit Operationen
- Algorithmen Suchen (rekursiv und iterativ), Minimum, Maximum
- Algorithmen Einfügen und Löschen
- Komplexität von binären Suchbäumen, ausgeglichene/entartete Bäume

### 8.4 Balancierte Suchbäume

- Balancieren von Suchbäumen
- Definition AVL-Bäume
- Operation Einfügen mit Rotation und Doppelrotation
- Operation Löschen

### 8.5 Suchen mit Hashtabellen

- Wörterbücher mit Operationen, Implementierungsbeispiele
- Adresstabellen
- Hashtabellen, Kollisionsauflösung mit Verkettung
- Hashfunktionen, Divisionsmethode, Multiplikationsmethode, universelles Hashing
- Kollisionsauflösung mit offener Adressierung

### 8.6 Suchen in Zeichenketten

- Brute-Force Search mit Komplexität
- Alphabet und Wörter, Präfix, Suffix und eigentlicher Rand
- Knuth-Morris-Pratt Algorithmus mit Komplexität

## 9 Graph-Algorithmen

### 9.1 Tiefensuche

- Bild als Feld, Segmentierung von Bildern mit Graphen
- Algorithmus Tiefensuche
- Berechnung von Spannwald/Spannbaum mit Tiefensuche
- Komplexität Tiefensuche
- Anwendungen von Tiefensuche

### 9.2 Breitensuche

- Algorithmus Breitensuche
- Komplexität und Eigenschaften von Breitensuche
- Anwendungen von Breitensuche

### 9.3 Kürzeste Pfade

- Gewicht von Pfaden, kürzeste Pfade
- Übersicht Algorithmen zur Berechnung von kürzesten Pfaden
- Gegenbeispiel für nicht-funktionierenden Greedy-Algorithmus für kürzeste Pfade
- Algorithmus Dijkstra
- Komplexität und Korrektheit von Dijkstra, Gegenbeispiel für negative Kanten
- Anwendungen von Dijkstra

### 9.4 Minimale Spannbäume

- Definition minimaler Spannbaum
- Übersicht Algorithmen zur Berechnung von minimalem Spannbaum
- Algorithmus Prim
- Komplexität Prim Algorithmus
- Anwendungen von Prim Algorithmus

## 10 Numerische Algorithmen

### 10.1 Matrizen

- Definition Matrix, Speicherung von Matrizen
- Matrix-Operationen
- Komplexität Matrizen-Multiplikation
- Strassen-Algorithmus zur Matrix-Multiplikation, Komplexität

### 10.2 Lineare Gleichungen

- Definition Lineare Gleichung, Lineares System
- Eigenschaften von linearen Systemen
- Inverse von Matrix, Invertierbarkeit von Matrizen
- Ansätze zur Lösung von linearen Systemen
- Gauss und Gauss-Jordan Elimination
- Günstige Matrixformen, Vorwärts- und Rückwärts-Substitution
- Matrix-Zerlegungen (Cholesky, QR, LUP, SVD)

### 10.3 Die LUP-Zerlegung

- LU-Zerlegung und Algorithmus
- Permutationsmatrizen
- LUP-Zerlegung und Algorithmus

## 10.4 Least squares Probleme

- Problemstellung Least squares, Wahl von Basisfunktionen
- Matrixnotation, Minimierung des Approximationsfehlers
- Least squares Lösung
- Normalengleichung
- Pseudoinverse, Berechnungsmethoden

## 10.5 Fast Fourier Transformation

- Fourier Transformation
- Polynome, Koordinationendarstellung
- Operationen mit Polynomen (Auswertung, Addition, Multiplikation)
- Stützstellendarstellung, Konversion mit Auswertung und Interpolation
- Algorithmen zur Interpolation
- Operationen mit Polynomen und Stützstellendarstellung (Addition, Multiplikation)
- Effiziente Multiplikation von Polynomen mit FFT/IFFT
- Komplexe Einheitswurzeln
- Diskrete Fourier Transformation
- Halbierungslemma, DFT mit Divide & Conquer
- rekursive FFT, Korrektheit und Laufzeit
- Interpolation mit komplexen Einheitswurzeln und IFFT
- Faltungstheorem
- Verbesserungen der rekursiven FFT
- Anwendung der FFT zur Filterung von Bildern

## 11 Datenkompression

### 11.1 Einführung

- Wichtigkeit und Beispiele von Datenkompression
- Verlustfreie und verlustbehaftete Kompression
- Codierungen und Kompressionsrate

### 11.2 Grundlagen Informationstheorie

- Eingabequelle mit Wahrscheinlichkeitsverteilung
- Informationsgehalt und Entropie
- Präfixcode und Kraft-Ungleichung
- Quellencodierungssatz von Shannon



### **11.3 Huffman Codes**

- Shannon-Fano Code
- Präfixcode als Baum
- Huffman Code
- Huffman in der Praxis

### **11.4 Lempel-Ziv Verfahren**

- Universelle Codierung mit Codebüchern
- LZ77 Funktionsweise, Beispiele, Ausblick auf LZSS
- LZ78 Funktionsweise, Beispiele, Ausblick auf LZW

### **11.5 JPEG**

- JPEG Kompressions-Schema
- Diskrete Cosinus Transformation
- Quantisierung
- Zig-Zag Schema, RLE und Huffman-Codierung

## **12 Kryptographie**

### **12.1 Einführung**

- Bedeutung und Begriffsklärung Kryptographie
- Informations-Sicherheit, Verschlüsselung und Angriffspunkte

### **12.2 Verfahren mit privaten Schlüsseln**

- Symmetrische Verschlüsselung: One Time Pads
- Verschlüsselungsstandards DES, 3DES, AES

### **12.3 Verfahren mit öffentlichen Schlüsseln**

- RSA Prinzip, Schlüsselerzeugung
- Modulare Exponentiation
- Angriffspunkte und Praxiseinsatz