



Vorlesungsinhalt

Semester: Wintersemester 2006/07
Vorlesung: Effiziente Algorithmen und Datenstrukturen I (4+2, 8 ECTS)
(mit Übungen)
Dozent: Prof. Dr. Ernst W. Mayr
Übungsleitung: Matthias Baumgart

Texte: Alfred V. Aho and John E. Hopcroft and Jeffrey D. Ullman:
“The design and analysis of computer algorithms”
Addison-Wesley Publishing Company: Reading (MA), 1974

Thomas H. Cormen and Charles E. Leiserson and R.L. Rivest
and Clifford Stein:
“Introduction to algorithms”
McGraw-Hill, 1990

Michael T. Goodrich and Roberto Tamassia:
“Algorithm design: Foundations, analysis, and internet examples”
John Wiley & Sons, 2002

Volker Heun:
“Grundlegende Algorithmen: Einführung in den Entwurf und die
Analyse effizienter Algorithmen”
2. Aufl., Vieweg: Braunschweig-Wiesbaden, 2003

Donald E. Knuth:
“The art of computer programming. V. 1: Fundamental algorithms”
3rd ed., Addison-Wesley Publishing Company: Reading (MA), 1997

Donald E. Knuth:
“The art of computer programming. V. 3: Sorting and searching”
3rd ed., Addison-Wesley Publishing Company: Reading (MA), 1997

Kurt Mehlhorn:
“Data structures and algorithms 1: Sorting and searching”
EATCS Monographs on Theoretical Computer Science
Springer-Verlag, 1984

Kurt Mehlhorn:
“Data structures and algorithms 2: Graph algorithms and
 \mathcal{NP} -Completeness”
EATCS Monographs on Theoretical Computer Science
Springer-Verlag, 1984

Christos H. Papadimitriou and Kenneth Steiglitz:
“Combinatorial optimization: Algorithms and complexity”
Prentice-Hall, Englewood Cliffs, NJ, 1982

Uwe Schöning:
“Algorithmik”
Spektrum Akademischer Verlag, 2001

Vorlesungsinhalt:

0. Organisatorisches

1. Vorlesungsinhalt
2. Literatur

I. Grundlagen

1. Ein einleitendes Beispiel
2. Versuch einer Definition
3. Ziel der Vorlesung
4. Maschinenmodelle
5. Komplexitätsmaße
6. Wachstumsverhalten von Funktionen
7. Rekursionsgleichungen
 - 7.1 Multiplikatoren
 - 7.2 Charakteristisches Polynom
 - 7.3 Erzeugendenfunktionen
 - 7.4 Transformation des Definitions- bzw. Wertebereichs

II. Höhere Datenstrukturen

1. Grundlegende Operationen
2. (Balancierte) Suchbäume
 - 2.1 (a, b) -Bäume
 - 2.2 Rot-Schwarz-Bäume
3. Binäre Suchbäume
 - 3.1 Natürliche binäre Suchbäume
 - 3.2 Höhenbalancierte binäre Suchbäume (AVL-Bäume)
 - 3.3 Gewichtsbalancierte Bäume
4. Hashing
 - 4.1 Grundlagen
 - 4.2 Methoden zur Kollisionsauflösung
 1. Geschlossene Hashverfahren (Chaining)
 2. Hashing mit offener Adressierung
 - 4.3 Universelles Hashing
 - 4.4 Perfektes Hashing
 1. Statisches perfektes Hashing

- 2. Dynamisches perfektes Hashing
- 5. Vorrangwarteschlangen - Priority Queues
 - 5.1 Binomial Queues (binomial heaps)
 - 5.2 Fibonacci-Heaps
 - 1. Die Datenstruktur
 - 2. Amortisierte Kostenanalyse für Fibonacci-Heaps
- 6. Sich selbst organisierende Datenstrukturen
 - 6.1 Motivation
 - 6.2 Sich selbst organisierende lineare Listen
 - 6.3 Sich selbst organisierende Binärbäume
 - 6.4 Splay-Trees als Suchbäume
 - 1. Die Splaying-Operation
 - 2. Amortisierte Kostenanalyse der Splay-Operation
 - 3. Wörterbuchoperationen in Splay-Trees
 - 6.5 Weitere Arten wichtiger Datenstrukturen
- 7. Radix-basierte Priority Queues
 - 7.1 Buckets
 - 1. 1-Level-Buckets
 - 2. 2-Level-Buckets
 - 3. k-Level-Buckets
 - 4. Literatur
 - 7.2 van Emde Boas-Priority Queues
 - 7.3 Radix-Heaps
- 8. Union/Find-Strukturen
 - 8.1 Motivation
 - 8.2 Union-Find-Datenstruktur
 - 1. Intrees
 - 2. Gewichtete Union (erste Verbesserung)
 - 3. Pfadkompression mit gewichteter Union (zweite Verbesserung)

III. Selektieren und Sortieren

- 1. Einleitung
- 2. Der Blum-Floyd-Pratt-Rivest-Tarjan Selektions-Algorithmus
- 3. Randomisierter Median-Algorithmus
- 4. Der Schönhage/Paterson/Pippenger-Median-Algorithmus
- 5. Eine untere Schranke für die Medianbestimmung

6. Eine bessere untere Schranke
7. Untere Schranke für (vergleichsbasiertes) Sortieren
8. Bucketsort im Schnitt
9. Quicksort

IV. Minimale Spannbäume

1. Grundlagen
2. Traversierung von Graphen
 - 2.1 DFS-Algorithmus
 - 2.2 BFS-Algorithmus
3. Minimale Spannbäume
 - 3.1 Konstruktion von minimalen Spannbäumen
 - 3.2 Generischer minimaler Spannbaum-Algorithmus
 - 3.3 Kruskal's Algorithmus
 - 3.4 Prim's Algorithmus
 - 3.5 Prim's Algorithmus, zweite Variante
 - 3.6 Erweiterungen

V. Kürzeste Pfade

1. Grundlegende Begriffe
2. Das single-source-shortest-path-Problem
 - 2.1 Dijkstra's Algorithmus
 - 2.2 Dijkstra's Algorithmus mit Radix-Heaps
 - 2.3 Bellman-Ford-Algorithmus
3. Floyd's Algorithmus für das all-pairs-shortest-path-Problem
4. Digraphen mit negativen Kanten
 - 4.1 Grundsätzliches
 - 4.2 Modifikation des Bellman-Ford-Algorithmus
 - 4.3 Modifikation des Floyd-Algorithmus
 - 4.4 Der Algorithmus von Johnson
5. Zusammenfassung
6. Transitiv Hülle
 - 6.1 Min-Plus-Matrix-Produkt und Min-Plus-Transitive Hülle
 - 6.2 Boolesche Matrixmultiplikation und Transitive Hülle
 - 6.3 Der 4-Russen-Algorithmus für Boolesche Matrixmultiplikation
 - 6.4 Transitive Hülle und DFS

7. Ein besserer Algorithmus für das all-pairs-shortest-distance (bzw. paths)-Problem in ungewichteten Digraphen
8. Ein Algorithmus für die transitive Hülle in Digraphen mit linearer erwarteter Laufzeit
9. Matrixmultiplikation á la Strassen

VI. Matchings in Graphen

1. Grundlagen
2. Kürzeste augmentierende Pfade
3. Maximum Matchings in bipartiten Graphen
4. Maximum Matchings in allgemeinen Graphen
5. Matchings in gewichteten bipartiten Graphen
 - 5.1 Zerlegung doppelt stochastischer Matrizen
 - 5.2 Matchings in knotengewichteten bipartiten Graphen
 - 5.3 Matchings in kantengewichteten bipartiten Graphen