

Rheinisch-Westfälische Technische Hochschule Aachen  
Lehrstuhl für Informatik VI

# **Algorithmen und Datenstrukturen**

Vorlesungsmitschrift zur Vorlesung im SS 2004

Prof. Dr.-Ing. H. Ney

Letzte Überarbeitung: 20. Juli 2004  
Thomas Deselaers

# Inhaltsverzeichnis

<b>1</b>	<b>Grundlagen</b>	<b>8</b>
1.1	Einführung	8
1.1.1	Problemstellungen	8
1.1.2	Aktualität des Themas	9
1.1.3	Ziele der Vorlesung	9
1.1.4	Hinweis auf das didaktische Problem der Vorlesung	9
1.1.5	Vorgehensweise	10
1.1.6	Datenstrukturen	10
1.2	Algorithmen und Komplexität	11
1.2.1	Random-Access-Machine (RAM)	13
1.2.2	Komplexitätsklassen	16
1.2.3	O-Notation für asymptotische Aussagen	17
1.2.4	Regeln für die Laufzeitanalyse	19
1.2.5	Beispiel: Fibonacci-Zahlen	21
1.2.6	Beispiel: Fakultät	24
1.3	Datenstrukturen	24
1.3.1	Datentypen	24
1.3.2	Listen	27
1.3.3	Stacks	37
1.3.4	Queues	39
1.3.5	Bäume	41
1.3.6	Abstrakte Datentypen	52
1.4	Entwurfsmethoden	54
1.4.1	Divide-and-Conquer-Strategie	54
1.4.2	Dynamische Programmierung	57
1.4.3	Memoization	60
1.5	Rekursionsgleichungen	61
1.5.1	Sukzessives Einsetzen	61
1.5.2	Master-Theorem für Rekursionsgleichungen	62
1.5.3	Rekursionsungleichungen	66

<b>2</b>	<b>Sortieren</b>	<b>68</b>
2.1	Einführung . . . . .	68
2.2	Elementare Sortierverfahren . . . . .	70
2.2.1	SelectionSort . . . . .	70
2.2.2	InsertionSort . . . . .	72
2.2.3	BubbleSort . . . . .	75
2.2.4	Indirektes Sortieren . . . . .	79
2.2.5	BucketSort . . . . .	81
2.3	QuickSort . . . . .	83
2.3.1	Beweis der Schranken von $\sum_{k=m}^n 1/k$ mittels Integral-Methode . . . . .	93
2.4	HeapSort . . . . .	94
2.4.1	Komplexitätsanalyse von HeapSort . . . . .	98
2.5	Untere und obere Schranken für das Sortierproblem . . . . .	100
2.6	Schranken für $n!$ . . . . .	103
2.7	MergeSort . . . . .	106
2.8	Zusammenfassung . . . . .	106
<b>3</b>	<b>Suchen in Mengen</b>	<b>107</b>
3.1	Problemstellung . . . . .	107
3.2	Einfache Implementierungen . . . . .	109
3.2.1	Ungeordnete Arrays und Listen . . . . .	109
3.2.2	Vergleichsbasierte Methoden . . . . .	109
3.2.3	Bitvektordarstellung (Kleines Universum) . . . . .	112
3.2.4	Spezielle Array-Implementierung . . . . .	113
3.3	Hashing . . . . .	116
3.3.1	Begriffe und Unterscheidung . . . . .	116
3.3.2	Hashfunktionen . . . . .	117
3.3.3	Wahrscheinlichkeit von Kollisionen . . . . .	119
3.3.4	Open Hashing (Hashing mit Verkettung) . . . . .	122
3.3.5	Closed Hashing (Hashing mit offener Adressierung) . . . . .	124
3.3.6	Zusammenfassung der Hashverfahren . . . . .	130
3.4	Binäre Suchbäume . . . . .	131
3.4.1	Allgemeine binäre Suchbäume . . . . .	131
3.4.2	Der optimale binäre Suchbaum . . . . .	144
3.5	Balancierte Bäume . . . . .	150
3.5.1	AVL-Bäume . . . . .	151
3.5.2	$(a,b)$ -Bäume . . . . .	156
3.6	Priority Queue und Heap . . . . .	158
<b>4</b>	<b>Graphen</b>	<b>160</b>
4.1	Motivation: Wozu braucht man Graphen? . . . . .	160
4.2	Definitionen und Graph-Darstellungen . . . . .	161
4.2.1	Graph-Darstellungen . . . . .	162
4.2.2	Programme zu den Darstellungen . . . . .	164

4.3	Graph-Durchlauf	165
4.3.1	Programm für den Graph-Durchlauf	166
4.4	Kürzeste Wege	171
4.4.1	Dijkstra-Algorithmus (Single-Source Best Path)	171
4.4.2	DAGs und kürzeste Wege	178
4.4.3	Floyd-Algorithmus (All-Pairs Best Path)	180
4.4.4	Warshall-Algorithmus	181
4.5	Minimaler Spannbaum	183
4.5.1	Definitionen	183
4.5.2	MST Property	183
4.5.3	Prim-Algorithmus	184
4.6	Zusammenfassung	187
<b>5</b>	<b>Ausgewählte Themen</b>	<b>188</b>
5.1	Dynamische Programmierung	188
5.1.1	Typische Vorgehensweise bei der dynamischen Programmierung	188
5.1.2	Rucksackproblem	188
5.1.3	Beispiele für dynamische Programmierung	192
5.2	Exact String Matching	193
5.2.1	Naiver Algorithmus (brute force)	193
5.2.2	Knuth-Morris-Pratt-Algorithmus (KMP-Algorithmus)	195
5.3	Approximate String Matching	200
5.3.1	Longest Common Subsequence	200
5.3.2	Shortest Common Supersequence	203
5.3.3	Edit Distance	205
5.3.4	Verfeinerungen der Editier-Distanz	208
5.4	Traveling Salesman Problem	211
5.4.1	Problemstellung	211
5.4.2	Exhaustive Search	212
5.4.3	Branch & Bound	213
5.4.4	Dynamische Programmierung	216
5.4.5	$A^*$	219

# Literaturverzeichnis

- [Cormen et al.] T.H. Cormen, C.E. Leiserson, R.L. Rivest: *Introduction to Algorithms*. MIT press / McGraw Hill, 10th printing, 1993.
- [Sedgewick 02] R. Sedgewick: *Algorithms in Java*, 3rd ed., Addison-Wesley, 2002.
- [Sedgewick 98] R. Sedgewick: *Algorithms*, 2nd ed., Addison-Wesley, 1998.
- [Sedgewick 88] R. Sedgewick: *Algorithms*. Addison-Wesley, 1988.
- [Sedgewick 93] R. Sedgewick: *Algorithms in Modula-3*. Addison-Wesley, 1993.
- [Schöning 97] U. Schöning: *Algorithmen – kurz gefasst*. Spektrum Akad. Verl., 1997, vergriffen.
- [Schöning 01] U. Schöning: *Algorithmik*. Spektrum Akad. Verl., 2001.
- [Güting] R.H. Güting: *Datenstrukturen und Algorithmen*. Teubner, 1992.
- [Aho et al. 83] A.V. Aho, J.E. Hopcroft, J.D. Ullman: *Data Structures and Algorithms*. Addison-Wesley, 1983.
- [Aho et al. 74] A.V. Aho, J.E. Hopcroft, J.D. Ullman: *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
- [Mehlhorn] K. Mehlhorn: *Datenstrukturen und effiziente Algorithmen*. Bde 1,2,3; (primär Band 1: „Sortieren und Suchen“), Teubner 1988. (Bde 2+3 vergriffen)
- [Wirth] N. Wirth: *Algorithmen und Datenstrukturen mit Modula-2*. 4. Auflage, Teubner 1986.
- [Aigner] M. Aigner: *Diskrete Mathematik*. Vieweg Studium, 1993.
- [Ottmann] T. Ottmann, P. Widmayer: *Algorithmen und Datenstrukturen*. 4. Auflage, Spektrum Akademischer Verlag, 2002