

*Leitfäden der Informatik*

**Ralf Hartmut Güting, Stefan Dieker**

# **Datenstrukturen und Algorithmen**

# *Leitfäden der Informatik*

Herausgegeben von

Prof. Dr. Bernd Becker  
Prof. Dr. Friedemann Mattern  
Prof. Dr. Heinrich Müller  
Prof. Dr. Wilhelm Schäfer  
Prof. Dr. Dorothea Wagner  
Prof. Dr. Ingo Wegener

Die Leitfäden der Informatik behandeln

- Themen aus der Theoretischen, Praktischen und Technischen Informatik entsprechend dem aktuellen Stand der Wissenschaft in einer systematischen und fundierten Darstellung des jeweiligen Gebietes.
- Methoden und Ergebnisse der Informatik, aufgearbeitet und dargestellt aus Sicht der Anwendungen in einer für Anwender verständlichen, exakten und präzisen Form.

Die Bände der Reihe wenden sich zum einen als Grundlage und Ergänzung zu Vorlesungen der Informatik an Studierende und Lehrende in Informatik-Studiengängen an Hochschulen, zum anderen an „Praktiker“, die sich einen Überblick über die Anwendungen der Informatik (-Methoden) verschaffen wollen; sie dienen aber auch in Wirtschaft, Industrie und Verwaltung tätigen Informatikern und Informatikerinnen zur Fortbildung in praxisrelevanten Fragestellungen ihres Faches.

**Ralf Hartmut Güting, Stefan Dieker**

# **Datenstrukturen und Algorithmen**

3., durchgesehene Auflage



B. G. Teubner Stuttgart · Leipzig · Wiesbaden

Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliografische Daten sind im Internet über <<http://dnb.ddb.de>> abrufbar.

### **Prof. Dr. rer. nat. Ralf Hartmut Güting**

Geboren 1955 in Lünen. Studium der Informatik mit Nebenfach Physik an der Universität Dortmund (1974/75 und 1977-1980). Wiss. Mitarbeiter an der Universität Dortmund (1981, 1982-1983). Einjähriger Forschungsaufenthalt an der McMaster University, Hamilton, Kanada bei Derik Wood (1981/82). Promotion 1983 bei Armin Cremers. Hochschulassistent an der Universität Dortmund (1984-1987). Einjähriger Forschungsaufenthalt am IBM Almaden Research Center, San Jose, Kalifornien, USA (1985). Professor für Informatik an der Universität Dortmund (1987-1989). Seit 1989 Professor für Praktische Informatik an der Fernuniversität Hagen.

### **Dr. rer. nat. Stefan Dieker**

Geboren 1968 in Düsseldorf. Studium der Angewandten Informatik mit den Nebenfächern Elektrotechnik und Betriebswirtschaftslehre (1989-1996). Softwareentwickler für betriebswirtschaftliche Standardsoftware (1996). Wissenschaftlicher Mitarbeiter an der Fernuniversität Hagen (1996-2001), Forschungsgebiet: Architektur und Implementierung erweiterbarer Datenbanksysteme. Promotion 2001 bei Ralf Hartmut Güting. Seit 2001 Anwendungsentwickler in der Industrie.

1. Auflage 1992

2., neu bearb. Auflage März 2003

3., durchgesehene Auflage Dezember 2004

Lektorat: Ulrich Sandten

Alle Rechte vorbehalten

© B. G. Teubner Verlag / GWV Fachverlage GmbH, Wiesbaden 2004

Der B. G. Teubner Verlag ist ein Unternehmen von Springer Science+Business Media.

[www.teubner.de](http://www.teubner.de)



Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Umschlaggestaltung: Ulrike Weigel, [www.CorporateDesignGroup.de](http://www.CorporateDesignGroup.de)

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier.

ISBN-13: 978-3-519-22121-0 e-ISBN-13: 978-3-322-89113-6

DOI: 10.1007/978-3-322-89113-6

*Für Edith, Nils David und Helge Jonathan*

*R.H.G.*

*Für Sabine, Tobias und Johannes*

*S.D.*

## Vorwort zur ersten Auflage

Effiziente Algorithmen und Datenstrukturen bilden ein zentrales Thema der Informatik. Wer programmiert, sollte zu den wichtigsten Problembereichen grundlegende Lösungsverfahren kennen; er sollte auch in der Lage sein, neue Algorithmen zu entwerfen, ggf. als Kombination bekannter Verfahren, und ihre Kosten in bezug auf Laufzeit und Speicherplatz zu analysieren. Datenstrukturen organisieren Information so, daß effiziente Algorithmen möglich werden. Dieses Buch möchte entsprechende Kenntnisse und Fähigkeiten vermitteln; es wendet sich vornehmlich an Studierende der Informatik im Grundstudium. Vorausgesetzt werden lediglich Grundkenntnisse der Programmierung, wie sie etwa durch Umgang mit einer Sprache wie PASCAL gegeben sind. Zum Verständnis werden zwar keine tiefgehenden mathematischen Vorkenntnisse, aber doch die Bereitschaft zum Umgang mit einfacher mathematischer Notation benötigt. Gelegentlich kommen bei der Analyse von Algorithmen auch etwas anspruchsvollere Berechnungen vor. Diese werden sorgfältig erklärt, und die benötigten Techniken werden im Rahmen des Buches eingeübt.

Grundlage für das Buch waren Vorlesungen zu Datenstrukturen und zu geometrischen Algorithmen, die ich an der Universität Dortmund gehalten habe; diese wurden später zu einem Kurs "Datenstrukturen" für die Fernuniversität Hagen ausgearbeitet und dort bereits einige Jahre eingesetzt. Der Stoffumfang des Buches umfaßt den einer einsemestrigen vierstündigen Vorlesung, die in Dortmund und Hagen jeweils im 3. Semester angeboten wird.

Es gibt zahlreiche Bücher zu Algorithmen und Datenstrukturen. Dieses Buch setzt folgende Akzente:

- Es wurde versucht, in relativ kompakter Form alle wichtigen Themenbereiche des Gebietes abzudecken. Dies erlaubt es, das Buch komplett als Vorlesungsgrundlage zu benutzen und darüber den Stoffumfang einer entsprechenden Prüfung zu definieren. Die meisten anderen Bücher sind wesentlich umfangreicher oder behandeln wichtige Themenbereiche (Graphalgorithmen, geometrische Algorithmen) nicht. Im ersten Fall hat ein Dozent die Möglichkeit, aber auch die Bürde, in erheblichem Maße auszuwählen.
- Die kompakte Darstellung wurde zum Teil erreicht durch Konzentration auf die Darstellung der wesentlichen Ideen. In diesem Buch wird die Darstellung von Algorithmen mit dem richtigen Abstraktionsgrad besonders betont. Die Idee eines Algorithmus kann auch in seitenlangen Programmen versteckt sein; dies sollte vermieden werden. Selbstverständlich geht die Beschreibung von Algorithmen immer Hand in Hand mit ihrer Analyse.

- *Datentyp* als Spezifikation und Anwendungssicht einer Datenstruktur und *Datenstruktur* als Implementierung eines Datentyps werden klar voneinander unterschieden. Es wird gezeigt, daß es zu einem Datentyp durchaus verschiedene Implementierungen geben kann. Die Spezifikation von Datentypen mit einer recht praxisnahen Methode wird an etlichen Beispielen vorgeführt.
- Die Forschung zu Algorithmen und Datenstrukturen ist in den letzten 15 Jahren nicht stehengeblieben. Vor allem die Behandlung geometrischer Objekte, also das Teilgebiet der *algorithmischen Geometrie*, hat in dieser Zeit einen stürmischen Aufschwung genommen. Dies wurde berücksichtigt, indem einerseits neue Ergebnisse zu klassischen Themen (etwa Sortieralgorithmen) mit aufgenommen wurden. Darüber hinaus setzt das Buch einen deutlichen Schwerpunkt im Bereich der algorithmischen Geometrie.

Das Kapitel zur algorithmischen Geometrie ist zweifellos etwas anspruchsvoller als der übrige Text. Es wird von Studenten gelegentlich als etwas schwierig, oft aber auch als hochinteressant empfunden. Ich bin der Meinung, daß die Beschäftigung mit diesem Kapitel sich aus mehreren Gründen besonders lohnt:

- Der Blick wird geweitet; man erkennt z.B., daß “schlichtes” Suchen und Sortieren nur der eindimensionale Spezialfall eines allgemeineren Problems ist, oder daß Bäume auch ganz anders konstruiert werden können als einfache binäre Suchbäume, daß man sie schachteln kann usw.
- Der Umgang mit *algorithmischen Paradigmen* wird anhand von *Plane-Sweep* und *Divide-and-Conquer* eingeübt; man sieht, daß man mit verschiedenen Techniken zu optimalen Lösungen für das gleiche Problem kommen kann.
- Der Entwurf von Algorithmen auf hohem Abstraktionsniveau zusammen mit systematischer Problemreduktion wird eingeübt.
- Schließlich begreift man, daß all die Algorithmen und Datenstrukturen der vorhergehenden Kapitel als *Werkzeuge* zur Konstruktion neuer Algorithmen eingesetzt werden können.

Man kann das gesamte Gebiet auf verschiedene Art gliedern, etwa nach Problembereichen und damit Algorithmen, nach Datentypen, oder nach Datenstrukturen. Dieses Buch führt in den Kapiteln 3 und 4 zunächst grundlegende Datentypen ein; danach “kippt” die Gliederung mit der Behandlung von Graphalgorithmen, Sortieralgorithmen und geometrischen Algorithmen auf die algorithmische Seite. Das letzte Kapitel behandelt *externes* Suchen und Sortieren. Ich fand es angemessener, B-Bäume und externes Mischsortieren getrennt zu behandeln, also nicht zusammen mit internen Baumstrukturen und Sortieren, da hier ein ganz anderes Kostenmaß zugrunde liegt.

Dem Text folgt ein kleiner Anhang, in dem die benötigten mathematischen Grundkenntnisse "importiert" werden. Fast jedes Kapitel schließt mit Aufgaben einfachen bis mittleren Schwierigkeitsgrades und Literaturhinweisen ab. Die Literaturhinweise beziehen sich oft auch auf Material, das den Stoff des Kapitels ergänzt oder weiterführt.

Wie schon erwähnt, ist das Buch vor allem als Grundlage oder Begleitmaterial für eine Vorlesung im Grundstudium gedacht. Darüber hinaus kann der Stoff des Kapitels über geometrische Algorithmen als Kern einer Spezialvorlesung im Hauptstudium benutzt werden.

Am Zustandekommen dieses Buches waren etliche Mitarbeiter, Studenten, Freunde und Kollegen beteiligt, denen ich herzlich danken möchte. Bernd Meyer und Gerd Westerman haben mitgeholfen, das Dortmunder Skript zu einem Kurs für die Fernuniversität umzuarbeiten; sie haben bei der Formulierung von Aufgaben auch eigene Entwürfe eingebracht. Auch Jürgen Freitag und Yan Liu haben Aufgaben mitentworfen. Für das Erfassen des Vorlesungstextes mit Zeichnungen, Setzen von Formeln usw. danke ich besonders Arno Aßmann. Er hat auch mit viel Mühe den Kurstext in das neue Buchformat umgesetzt und bei all diesen Arbeiten großes Geschick und Verständnis bewiesen. Viele Studenten der Universität Dortmund und vor allem der Fernuniversität haben Korrekturen und Verbesserungsvorschläge zu früheren Versionen des Textes eingebracht. Wegen ihres besonderen Einsatzes dabei möchte ich Michael Kohler, Ingrid Nagel und Matthias Wolpers namentlich erwähnen. Teile der Buchversion wurden von Arno Aßmann, Martin Erwig, Bernd Meyer, Markus Schneider und Gerd Westerman gelesen und kommentiert; ich danke ihnen für die oft sehr detaillierten Korrekturen und Verbesserungsvorschläge.

Für die Unterstützung des Buchprojekts und ihre Ermutigung danke ich herzlich den Herausgebern der Reihe "Leitfäden und Monographien der Informatik", Hans-Jürgen Appelrath und Volker Claus. Auch sie haben das Manuskript gelesen und sehr nützliche Kommentare und Verbesserungsvorschläge eingebracht. Dem Teubner-Verlag danke ich für die reibungslose Zusammenarbeit.

Trotz aller Unterstützung gehen verbleibende Fehler und Schwächen der Darstellung natürlich zu meinen Lasten. Ich bitte Sie, die Leser des Buches, mich darauf aufmerksam zu machen.

Hagen, im Februar 1992

Ralf Hartmut Güting



## Vorwort zur zweiten Auflage

Seit dem Erscheinen der ersten Auflage hat sich das Buch als Basis- oder Sekundärliteratur zur Vorlesung *Datenstrukturen und Algorithmen* im Grundstudium der Informatik an vielen Hochschulen bewährt. In Form des gleichnamigen Kurses ist es auch an der Fernuniversität Hagen alljährlich erfolgreich im Einsatz. Zahlreiche konstruktive Kommentare engagierter Studierender sind seitdem in Details des Kurstextes eingeflossen.

Das “gereifte” Kurs-Manuskript bildet die Grundlage für die vorliegende zweite, überarbeitete Auflage. Die Algorithmen und die Datenstrukturen der algorithmischen Ebene sind dieselben geblieben. Natürlich gibt es auch weiterhin immer wieder neue Ergebnisse zu Algorithmen und Datenstrukturen; bei beschränktem Umfang halten wir es aber nicht für gerechtfertigt, Material der hier zu legenden Grundlagen zu verdrängen. Wir haben jedoch die Präsentation dieser Konzepte und, als augenfälligste Veränderung, die Implementierungsbeispiele der Entwicklung der letzten Jahre angepasst. Dazu wurden im wesentlichen folgende Erneuerungen vorgenommen:

- Modula-2 als Programmiersprache für Implementierungsbeispiele wurde von Java abgelöst. 1992 war Modula-2 die Sprache der Wahl in der Informatikausbildung und wurde, wie das sehr ähnliche Pascal, auch in der industriellen Praxis häufig verwendet. Im Zuge der Entwicklung objektorientierter Programmiersprachen ist heute Java an diese Stelle getreten.
- Eine wichtige Rolle in der alten wie der neuen Version des Buches spielt die Betonung des Unterschiedes zwischen *Programmen*, die in einer konkreten Programmiersprache formuliert sind, und *Algorithmen*, die von konkreten Programmiersprachen abstrahieren und neben programmiersprachlichen Konstrukten auch mathematische Notationen und natürlichsprachliche Formulierungen enthalten. In der neuen Auflage wird der Unterschied zwischen Algorithmen und Programmen noch deutlicher herausgestellt.
- Die Notation von Algorithmen wurde an heutige Standards angepaßt.

Gemäß seinem Ursprung als Kurs der FernUniversität ist dieses Buch hervorragend zum Selbststudium geeignet. Die zweite Auflage unterstützt diesen Anspruch dadurch, daß zum Selbsttest gedachte Aufgaben in den Text eingestreut sind, deren Lösungen im Anhang angegeben sind. Wir empfehlen dem Leser, die Selbsttestaufgaben möglichst zu bearbeiten, also nicht einfach die Lösung nachzusehen. Für ein eingehendes Verständnis des Stoffes ist der eigene kreative Umgang mit den gestellten Problemen von entscheidender Bedeutung.

Am Ende fast jeden Kapitels finden sich weitere Aufgaben, die z.B. von Dozenten in der Lehre genutzt werden können. Natürlich kann sich auch der eifrige Leser daran versuchen! Zu diesen Aufgaben sind keine Lösungen angegeben.

Wir wünschen Ihnen viel Freude und Erfolg bei der Arbeit an dem nicht immer einfachen, aber hochinteressanten und vielfältigen Thema *Datenstrukturen und Algorithmen*.

Hagen, im Februar 2003

Ralf Hartmut Güting

Stefan Dieker

### **Vorwort zur dritten Auflage**

Wir haben uns über die positiven Reaktionen vieler Leser auf die zweite Auflage des Buches sehr gefreut. Vor allem die Umstellung der Programmbeispiele auf Java hat großen Anklang gefunden. Dementsprechend war die zweite Auflage schnell verkauft.

In der nun vorliegenden dritten Auflage wurden eine Reihe von Druckfehlern und wenige kleine technische Fehler korrigiert. Wir danken allen Lesern, die uns auf solche Fehler hingewiesen haben.

Hagen, im November 2004

Ralf Hartmut Güting

Stefan Dieker

# Inhalt

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Einführung</b>  | <b>1</b>  |
| 1.1      | Algorithmen und ihre Analyse                               | 2         |
| 1.2      | Datenstrukturen, Algebren, Abstrakte Datentypen            | 22        |
| 1.3      | Grundbegriffe  | 32        |
| 1.4      | Weitere Aufgaben   | 35        |
| 1.5      | Literaturhinweise  | 36        |
| <b>2</b> | <b>Programmiersprachliche Konzepte für Datenstrukturen</b> | <b>39</b> |
| 2.1      | Datentypen in Java   | 40        |
| 2.1.1    | Basisdatentypen  | 41        |
| 2.1.2    | Arrays   | 42        |
| 2.1.3    | Klassen  | 45        |
| 2.2      | Dynamische Datenstrukturen                                 | 49        |
| 2.2.1    | Programmiersprachenunabhängig: Zeigertypen                 | 49        |
| 2.2.2    | Zeiger in Java: Referenztypen                              | 53        |
| 2.3      | Weitere Konzepte zur Konstruktion von Datentypen           | 57        |
|          | Aufzählungstypen   | 58        |
|          | Unterbereichstypen   | 59        |
|          | Sets   | 59        |
| 2.4      | Literaturhinweise  | 61        |
| <b>3</b> | <b>Grundlegende Datentypen</b>                             | <b>63</b> |
| 3.1      | Sequenzen (Folgen, Listen)                                 | 63        |
| 3.1.1    | Modelle  | 64        |
|          | (a) Listen mit first, rest, append, concat                 | 64        |
|          | (b) Listen mit expliziten Positionen                       | 65        |
| 3.1.2    | Implementierungen  | 68        |
|          | (a) Doppelt verkettete Liste                               | 68        |
|          | (b) Einfach verkettete Liste                               | 73        |
|          | (c) Sequentielle Darstellung im Array                      | 77        |
|          | (d) Einfach oder doppelt verkettete Liste im Array         | 78        |
| 3.2      | Stacks   | 82        |
| 3.3      | Queues   | 89        |
| 3.4      | Abbildungen  | 91        |
| 3.5      | Binäre Bäume   | 92        |
|          | Implementierungen  | 99        |
|          | (a) mit Zeigern  | 99        |
|          | (b) Array - Einbettung                                     | 100       |

|          |   |            |
|----------|---|------------|
| 3.6      | (Allgemeine) Bäume                              | 101        |
|          | Implementierungen                               | 104        |
|          | (a) über Arrays                                 | 104        |
|          | (b) über Binärbäume                             | 104        |
| 3.7      | Weitere Aufgaben                                | 105        |
| 3.8      | Literaturhinweise                               | 107        |
| <b>4</b> | <b>Datentypen zur Darstellung von Mengen</b>    | <b>109</b> |
| 4.1      | Mengen mit Durchschnitt, Vereinigung, Differenz | 109        |
|          | Implementierungen                               | 110        |
|          | (a) Bitvektor                                   | 110        |
|          | (b) Ungeordnete Liste                           | 111        |
|          | (c) Geordnete Liste                             | 111        |
| 4.2      | Dictionaries: Mengen mit INSERT, DELETE, MEMBER | 113        |
| 4.2.1    | Einfache Implementierungen                      | 114        |
| 4.2.2    | Hashing   | 115        |
|          | Analyse des "idealen" geschlossenen Hashing     | 120        |
|          | Kollisionsstrategien                            | 126        |
|          | (a) Lineares Sondieren (Verallgemeinerung)      | 126        |
|          | (b) Quadratisches Sondieren                     | 126        |
|          | (c) Doppel-Hashing                              | 127        |
|          | Hashfunktionen                                  | 128        |
|          | (a) Divisionsmethode                            | 128        |
|          | (b) Mittel-Quadrat-Methode                      | 128        |
| 4.2.3    | Binäre Suchbäume                                | 129        |
|          | Durchschnittsanalyse für binäre Suchbäume       | 136        |
| 4.2.4    | AVL-Bäume                                       | 141        |
|          | Updates   | 141        |
|          | Rebalancieren                                   | 142        |
| 4.3      | Priority Queues: Mengen mit INSERT, DELETETEMIN | 152        |
|          | Implementierung                                 | 153        |
| 4.4      | Partitionen von Mengen mit MERGE, FIND          | 156        |
|          | Implementierungen                               | 157        |
|          | (a) Implementierung mit Arrays                  | 157        |
|          | (b) Implementierung mit Bäumen                  | 160        |
|          | Letzte Verbesserung: Pfadkompression            | 162        |
| 4.5      | Weitere Aufgaben                                | 163        |
| 4.6      | Literaturhinweise                               | 166        |

|          |  |            |
|----------|--|------------|
| <b>5</b> | <b>Graphen und Graph-Algorithmen</b>                             | <b>167</b> |
| 5.1      | Gerichtete Graphen   | 168        |
| 5.2      | (Speicher-) Darstellungen von Graphen                            | 170        |
|          | (a) Adjazenzmatrix   | 171        |
|          | (b) Adjazenzlisten   | 172        |
| 5.3      | Graphdurchlauf   | 173        |
| 5.4      | Bestimmung kürzester Wege von einem Knoten zu allen anderen      | 177        |
|          | Implementierungen des Algorithmus Dijkstra                       | 183        |
|          | (a) mit einer Adjazenzmatrix                                     | 183        |
|          | (b) mit Adjazenzlisten und als Heap dargestellter Priority Queue | 183        |
| 5.5      | Bestimmung kürzester Wege zwischen allen Knoten im Graphen       | 184        |
|          | Implementierung des Algorithmus von Floyd                        | 187        |
|          | (a) mit der Kostenmatrix-Darstellung                             | 187        |
|          | (b) mit Adjazenzlisten   | 188        |
| 5.6      | Transitive Hülle   | 189        |
| 5.7      | Starke Komponenten   | 190        |
| 5.8      | Ungerichtete Graphen   | 194        |
| 5.9      | Minimaler Spannbaum (Algorithmus von Kruskal)                    | 195        |
| 5.10     | Weitere Aufgaben   | 198        |
| 5.11     | Literaturhinweise  | 201        |
| <b>6</b> | <b>Sortieralgorithmen</b>  | <b>203</b> |
| 6.1      | Einfache Sortierverfahren: Direktes Auswählen und Einfügen       | 204        |
| 6.2      | Divide-and-Conquer-Methoden: Mergesort und Quicksort             | 207        |
|          | Durchschnittsanalyse für Quicksort                               | 215        |
| 6.3      | Verfeinertes Auswählen und Einfügen: Heapsort und Baumsortieren  | 218        |
|          | Standard-Heapsort  | 218        |
|          | Analyse von Heapsort   | 220        |
|          | Bottom-Up-Heapsort   | 222        |
| 6.4      | Untere Schranke für allgemeine Sortierverfahren                  | 224        |
| 6.5      | Sortieren durch Fachverteilen: Bucketsort und Radixsort          | 228        |
| 6.6      | Weitere Aufgaben   | 231        |
| 6.7      | Literaturhinweise  | 232        |

|          |  |            |
|----------|--|------------|
| <b>7</b> | <b>Geometrische Algorithmen</b>                              | <b>235</b> |
| 7.1      | Plane-Sweep-Algorithmen für orthogonale Objekte in der Ebene | 240        |
| 7.1.1    | Das Segmentschnitt-Problem                                   | 240        |
| 7.1.2    | Das Rechteckschnitt-Problem                                  | 245        |
|          | Das Punkteinschluß-Problem und seine Plane-Sweep-Reduktion   | 246        |
|          | Der Segment-Baum   | 248        |
|          | Komplexität der Lösungen                                     | 250        |
| 7.1.3    | Das Maßproblem   | 252        |
|          | Plane-Sweep-Reduktion  | 252        |
|          | Ein modifizierter Segment-Baum                               | 254        |
|          | Komplexität der Lösung des Maßproblems                       | 255        |
| 7.2      | Divide-and-Conquer-Algorithmen für orthogonale Objekte       | 256        |
| 7.2.1    | Das Segmentschnitt-Problem                                   | 257        |
| 7.2.2    | Das Maßproblem   | 263        |
| 7.2.3    | Das Konturproblem  | 270        |
| 7.3      | Suchen auf Mengen orthogonaler Objekte                       | 276        |
|          | Der Range-Baum   | 277        |
|          | Der Intervall-Baum   | 278        |
|          | Baumhierarchien  | 282        |
| 7.4      | Plane-Sweep-Algorithmen für beliebig orientierte Objekte     | 285        |
| 7.5      | Weitere Aufgaben   | 288        |
| 7.6      | Literaturhinweise  | 291        |
| <b>8</b> | <b>Externes Suchen und Sortieren</b>                         | <b>295</b> |
| 8.1      | Externes Suchen: B-Bäume                                     | 296        |
|          | Einfügen und Löschen   | 300        |
|          | Overflow   | 301        |
|          | Underflow  | 302        |
| 8.2      | Externes Sortieren   | 306        |
|          | Anfangsläufe fester Länge - direktes Mischen                 | 309        |
|          | Anfangsläufe variabler Länge - natürliches Mischen           | 310        |
|          | Vielweg-Mischen  | 312        |
| 8.3      | Weitere Aufgaben   | 313        |
| 8.4      | Literaturhinweise  | 315        |
|          | <b>Mathematische Grundlagen</b>                              | <b>317</b> |
|          | <b>Lösungen zu den Selbsttestaufgaben</b>                    | <b>325</b> |
|          | <b>Literatur</b>   | <b>361</b> |
|          | <b>Index</b>   | <b>371</b> |