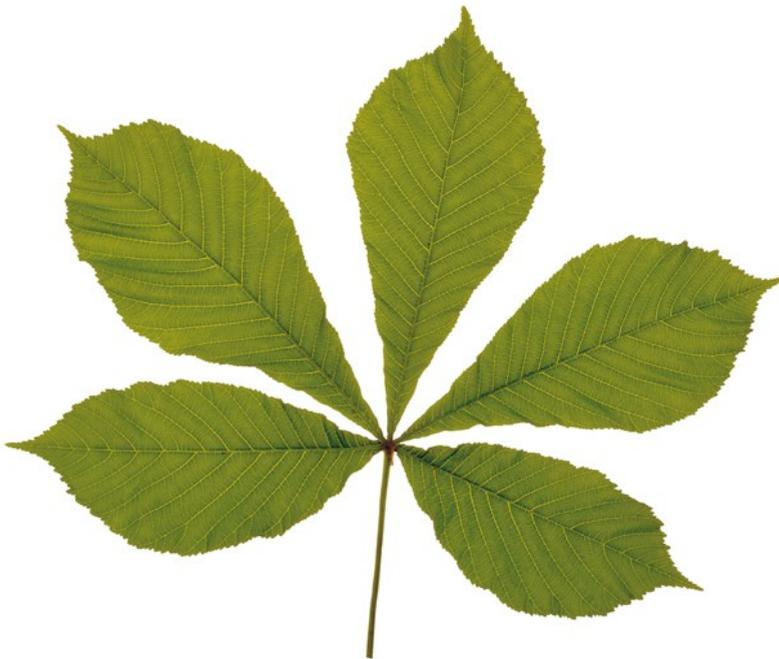




Th. H. Cormen | Ch. E. Leiserson | R. Rivest | C. Stein

Algorithmen – Eine Einführung

3. Auflage



Oldenbourg



Algorithmen – Eine Einführung

von

Prof. Dr. Thomas H. Cormen,
Prof. Dr. Charles E. Leiserson,
Prof. Dr. Ronald Rivest,
Prof. Dr. Clifford Stein

Aus dem Englischen von
Prof. Dr. rer. nat. habil. Paul Molitor,
Martin-Luther-Universität Halle-Wittenberg

3., überarbeitete und erweiterte Auflage

Oldenbourg Verlag München

Autorisierte Übersetzung der englischsprachigen Ausgabe, die bei

*The MIT Press, Massachusetts Institute of Technology und McGraw-Hill Book
Company*

unter dem Titel

Introduction to Algorithms, Third Edition

erschienen ist.

Copyright © 2009 MIT Press, McGraw-Hill Book Company

Übersetzung:

Prof. Dr. rer. nat. habil. Paul Molitor, Institut für Informatik, Martin-Luther-Universität Halle-Wittenberg.

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© 2010 Oldenbourg Wissenschaftsverlag GmbH
Rosenheimer Straße 145, D-81671 München
Telefon: (089) 45051-0
oldenbourg.de

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

Lektorat: Kathrin Mönch
Herstellung: Anna Grosser
Coverentwurf: Kochan & Partner, München
Gedruckt auf säure- und chlorfreiem Papier
Gesamtherstellung: Druckhaus „Thomas Müntzer“ GmbH, Bad Langensalza

ISBN 978-3-486-59002-9

Inhaltsverzeichnis

Vorwort	XIII
I Grundlagen	1
1 Die Rolle von Algorithmen in der elektronischen Datenverarbeitung	5
1.1 Algorithmen	5
1.2 Algorithmen als Technologie	11
2 Ein einführendes Beispiel	17
2.1 Sortieren durch Einfügen	17
2.2 Analyse von Algorithmen	24
2.3 Entwurf von Algorithmen	30
3 Wachstum von Funktionen	45
3.1 Asymptotische Notation	45
3.2 Standardnotationen und Standardfunktionen	55
4 Teile-und-Beherrsche	67
4.1 Das Max-Teilfeld-Problem	70
4.2 Strassens Algorithmus zur Matrizenmultiplikation	77
4.3 Die Substitutionsmethode zum Lösen von Rekursionsgleichungen	85
4.4 Die Rekursionsbaum-Methode zum Lösen von Rekursionsgleichungen ..	89
4.5 Die Mastermethode zum Lösen von Rekursionsgleichungen	95
4.6 *Beweis des Mastertheorems	99
5 Probabilistische Analyse und randomisierte Algorithmen	115
5.1 Das Bewerberproblem	115
5.2 Indikatorfunktionen	118

5.3	Randomisierte Algorithmen	123
5.4	* Probabilistische Analyse und mehr zur Verwendung der Indikatorfunktion	130
II	Sortieren und Ranggrößen	147
6	Heapsort	153
6.1	Heaps	153
6.2	Die Heap-Eigenschaft aufrechterhalten	156
6.3	Einen Heap bauen	158
6.4	Der Heapsort-Algorithmus	161
6.5	Prioritätswarteschlangen	162
7	Quicksort	171
7.1	Beschreibung von Quicksort	171
7.2	Die Performanz von Quicksort	175
7.3	Eine randomisierte Version von Quicksort	179
7.4	Analyse von Quicksort	181
8	Sortieren in linearer Zeit	191
8.1	Untere Schranken für das Sortieren	191
8.2	Countingsort	194
8.3	Radixsort	197
8.4	Bucketsort	200
9	Mediane und Ranggrößen	213
9.1	Minimum und Maximum	213
9.2	Auswahl in linearer erwarteter Zeit	215
9.3	Auswahl in linearer Zeit im schlechtesten Fall	219
III	Datenstrukturen	227
10	Elementare Datenstrukturen	233
10.1	Stapel und Warteschlangen	233
10.2	Verkettete Listen	237

10.3 Implementierung von Zeigern und Objekten	242
10.4 Darstellung von gerichteten Bäumen	246
11 Hashtabellen	255
11.1 Adresstabellen mit direktem Zugriff	256
11.2 Hashtabellen	258
11.3 Hashfunktionen	264
11.4 Offene Adressierung	272
11.5 * Perfektes Hashing	280
12 Binäre Suchbäume	289
12.1 Was ist ein binärer Suchbaum?	289
12.2 Abfragen in einem binären Suchbaum	292
12.3 Einfügen und Löschen	296
12.4 * Zufällig erzeugte binäre Suchbäume	302
13 Rot-Schwarz-Bäume	311
13.1 Eigenschaften von Rot-Schwarz-Bäumen	311
13.2 Rotationen	315
13.3 Einfügen eines Knotens	317
13.4 Löschen eines Knotens	325
14 Erweitern von Datenstrukturen	341
14.1 Dynamische Ranggröße	341
14.2 Wie man eine Datenstruktur erweitert	347
14.3 Intervallbäume	350
IV Fortgeschrittene Entwurfs- und Analysetechniken	359
15 Dynamische Programmierung	363
15.1 Schneiden von Eisenstangen	364
15.2 Matrizen-Kettenmultiplikation	374
15.3 Elemente dynamischer Programmierung	381
15.4 Längste gemeinsame Teilsequenz	393
15.5 Optimale binäre Suchbäume	399

16 Greedy-Algorithmen	417
16.1 Ein Aktivitäten-Auswahl-Problem	418
16.2 Elemente der Greedy-Strategie.....	425
16.3 Huffman-Codierungen	431
16.4 * Matroiden und Greedy-Methoden	440
16.5 * Ein Task-Scheduling-Problem als Matroid	447
17 Amortisierte Analyse	455
17.1 Aggregat-Analyse	456
17.2 Account-Methode.....	460
17.3 Die Potentialmethode.....	462
17.4 Dynamische Tabellen	466
V Höhere Datenstrukturen	483
18 B-Bäume	489
18.1 Die Definition von B-Bäumen.....	493
18.2 Grundoperationen auf B-Bäumen	496
18.3 Löschen eines Schlüssels aus einem B-Baum	504
19 Fibonacci-Heaps	511
19.1 Die Struktur von Fibonacci-Heaps	513
19.2 Operationen der fusionierbaren Heaps.....	516
19.3 Verringern eines Schlüssels und Entfernen eines Knotens	525
19.4 Beschränkung des maximalen Grades	529
20 van-Emde-Boas-Bäume	539
20.1 Vorbereitende Ansätze.....	540
20.2 Eine rekursive Datenstruktur	544
20.3 Die van-Emde-Boas-Bäume	553
21 Datenstrukturen disjunkter Mengen	569
21.1 Operationen auf disjunkten Mengen	569
21.2 Darstellung disjunkter Mengen mithilfe verketteter Listen	572

21.3 Wälder disjunkter Mengen	576
21.4 * Analyse der Vereinigung nach dem Rang mit Pfadverkürzung	580
VI Graphenalgorithmien	595
22 Elementare Graphenalgorithmien	599
22.1 Darstellungen von Graphen	599
22.2 Breitensuche	603
22.3 Tiefensuche	613
22.4 Topologisches Sortieren	622
22.5 Starke Zusammenhangskomponenten.....	626
23 Minimale Spannbäume	635
23.1 Aufbau eines minimalen Spannbaums	636
23.2 Die Algorithmen von Kruskal und Prim.....	641
24 Kürzeste Pfade von einem Startknoten aus	655
24.1 Der Bellman-Ford-Algorithmus	663
24.2 Kürzeste Pfade von einem Startknoten aus in DAGs	667
24.3 Dijkstras Algorithmus	670
24.4 Differenzbedingungen und kürzeste Pfade	677
24.5 Beweise der Eigenschaften kürzester Pfade	683
25 Kürzeste Pfade für alle Knotenpaare	697
25.1 Kürzeste Pfade und Matrizenmultiplikation	699
25.2 Der Floyd-Warshall-Algorithmus.....	705
25.3 Johnsons Algorithmus für dünn besetzte Graphen	713
26 Maximaler Fluss	721
26.1 Flussnetzwerke	722
26.2 Die Ford-Fulkerson-Methode	727
26.3 Maximales bipartites Matching	745
26.4 * Push/Relabel-Algorithmen	749
26.5 * Der Relabel-to-Front-Algorithmus	762

VII	Ausgewählte Themen	781
27	Mehrfädige Algorithmen	785
27.1	Grundlagen von dynamischem Multithreading.....	787
27.2	Mehrfädige Matrizenmultiplikation.....	806
27.3	Mehrfädiges Sortieren durch Mischen	811
28	Operationen auf Matrizen	827
28.1	Lösen linearer Gleichungssysteme	827
28.2	Matrixinversion	841
28.3	Symmetrische positiv definite Matrizen, Summe der quadratischen Fehler	846
29	Lineare Programmierung	857
29.1	Standard- und Schlupfform	864
29.2	Darstellung von Problemen als lineare Programme	872
29.3	Der Simplexalgorithmus	878
29.4	Dualität	893
29.5	Die initiale zulässige Basislösung	899
30	Polynome und die FFT	911
30.1	Darstellung von Polynomen	913
30.2	Die DFT und FFT	919
30.3	Effiziente Implementierung der FFT	927
31	Zahlentheoretische Algorithmen	937
31.1	Elementare zahlentheoretische Begriffe	938
31.2	Größter gemeinsamer Teiler	944
31.3	Modulare Arithmetik	950
31.4	Lösen modularer linearer Gleichungen.....	957
31.5	Der chinesische Restsatz	962
31.6	Potenzen eines Elements	965
31.7	Das RSA-Kryptosystem	970
31.8	* Primzahltests	977
31.9	* Primfaktorzerlegung	987

32 String-Matching	997
32.1 Der naive String-Matching-Algorithmus	999
32.2 Der Rabin-Karp-Algorithmus	1002
32.3 String-Matching mit endlichen Automaten	1007
32.4 * Der Knuth-Morris-Pratt-Algorithmus	1014
33 Algorithmische Geometrie	1025
33.1 Eigenschaften von Strecken	1026
33.2 Bestimmung von Schnittpunkten in einer Menge von Strecken	1032
33.3 Bestimmen der konvexen Hülle	1039
33.4 Berechnung des dichtesten Punktepaars	1050
34 NP-Vollständigkeit	1059
34.1 Polynomielle Zeit	1064
34.2 Verifikation in polynomieller Zeit	1072
34.3 NP-Vollständigkeit und Reduktion	1077
34.4 NP-Vollständigkeitsbeweise	1088
34.5 NP-vollständige Probleme	1096
35 Approximationsalgorithmen	1117
35.1 Das Knotenüberdeckungsproblem	1119
35.2 Das Problem des Handelsreisenden	1122
35.3 Das Mengenüberdeckungsproblem	1128
35.4 Randomisierung und lineare Programmierung	1134
35.5 Das Teilsommenproblem	1139
VIII Anhang	1151
A Summen	1155
A.1 Summenformeln und Eigenschaften	1155
A.2 Abschätzungen für Summen	1159
B Mengen usw.	1169
B.1 Mengen	1169
B.2 Relationen	1174

B.3 Funktionen	1176
B.4 Graphen	1178
B.5 Bäume	1183
C Kombinatorik und Wahrscheinlichkeitstheorie	1193
C.1 Kombinatorik	1193
C.2 Wahrscheinlichkeiten	1199
C.3 Diskrete Zufallsvariablen.....	1205
C.4 Die geometrische Verteilung und die Binomialverteilung	1211
C.5 * Die Ränder der Binomialverteilung	1217
D Matrizen	1227
D.1 Matrizen und Matrizenoperationen	1227
D.2 Elementare Matrizeneigenschaften	1232
Literaturverzeichnis	1241
Index	1265

Vorwort

Bevor es Rechner gab, gab es Algorithmen. Aber jetzt, wo es Rechner gibt, gibt es sogar noch mehr Algorithmen. Die elektronische Datenverarbeitung beruht auf Algorithmen.

Dieses Buch bietet eine umfassende Einführung in das moderne Studium von Algorithmen. Es stellt viele Algorithmen vor, behandelt sie detailliert und macht zudem deren Entwurf und deren Analyse allen Leserschichten zugänglich. Wir haben uns bemüht, Erklärungen elementar zu halten, ohne auf Tiefe oder mathematische Exaktheit zu verzichten.

Jedes Kapitel stellt einen Algorithmus, eine Entwurfstechnik, ein Anwendungsgebiet oder ein verwandtes Thema vor. Algorithmen werden in deutscher Sprache beschrieben und in Pseudocode entworfen, der für jeden lesbar sein sollte, der schon selbst ein wenig programmiert hat. Das Buch enthält 244 Abbildungen – viele bestehen aus mehreren Teilen – die verdeutlichen, wie die Algorithmen arbeiten. Da wir *Effizienz* als Entwurfskriterium betonen, schließen wir eine sorgfältige Analyse der Laufzeiten all unserer Programme mit ein.

Der Text ist in erster Linie für den Gebrauch in Grundvorlesungen und graduierten Lehrveranstaltungen zu Algorithmen und Datenstrukturen gedacht. Da er sowohl technische Fragen als auch mathematische Aspekte des Algorithmenentwurfs diskutiert, ist er ebenso zum Selbststudium für Menschen mit technischen Berufen geeignet.

In dieser dritten Auflage haben wir abermals das gesamte Buch aktualisiert. Die Änderungen sind vielfältig und umfassen insbesondere neue Kapitel, überarbeiteter Pseudocode und einen lebhafteren Schreibstil.¹

An die Lehrenden

Wir haben das Buch so gestaltet, dass es sowohl vielseitig als auch vollständig ist. Es sollte für Sie für eine Vielzahl von Lehrveranstaltungen von Nutzen sein, sowohl in einer Lehrveranstaltung zu Datenstrukturen für Studienanfänger als auch in einem Graduiertenkurs über Algorithmen. Da wir erheblich mehr Stoff bereitgestellt haben als eine typische, einsemestrige Veranstaltung umfasst, können Sie dieses Buch als eine Art „Buffet“ oder „bunte Mischung“ betrachten, aus der Sie denjenigen Stoff auswählen und entnehmen können, der die Lehrveranstaltung, die Sie halten möchten, am besten unterstützt.

¹Die vorliegende deutsche Übersetzung enthält nicht nur die in der dritten Auflage des englischen Buches enthaltenen Änderungen. Wir haben auch die alte deutsche Auflage auf umständliche Übersetzungen durchforstet und sie entsprechend überarbeitet.

Es sollte Ihnen leicht fallen, Ihre Lehrveranstaltung einfach aus den Kapiteln aufzubauen, die sie benötigen. Wir haben die Kapitel relativ eigenständig gestaltet, sodass Sie sich keine Gedanken über unerwartete und unnötige Abhängigkeiten eines Kapitels gegenüber einem anderen machen müssen. Jedes Kapitel stellt zuerst den einfacheren Stoff bereit, später dann die komplizierteren Sachverhalte, wobei Abschnitte logische Zusammenhänge markieren. Innerhalb einer Lehrveranstaltung für Studienanfänger werden Sie möglicherweise nur die ersten Abschnitte eines Kapitels verwenden; innerhalb einer Graduiertenveranstaltung könnten Sie das gesamte Kapitel behandeln.

Wir haben insgesamt 957 Übungen und 158 Problemstellungen eingebunden. Jeder Abschnitt endet mit Übungen, Kapitel schließen mit Problemstellungen ab. Die Übungen sind im Allgemeinen kurze Fragen, die das Beherrschen des Lehrstoffs testen. Einige sind einfache, als Selbsttest gedachte Aufgaben, während andere wesentlich umfangreicher und als Hausarbeiten geeignet sind. Die Problemstellungen sind aufwendigere Fallstudien, die häufig neuen Stoff einführen; sie bestehen häufig aus mehreren Fragen, die die Studierenden durch die zum Erhalt der Lösung notwendigen Schritte führen.

Aus der Erfahrung mit den vorherigen Auflagen dieses Buches haben wir Lösungen zu einigen, aber beileibe nicht allen Problemstellungen und Übungen öffentlich zugänglich gemacht. Sie können über unsere Webseite <http://mitpress.mit.edu/algorithms> auf diese zugreifen. Sie sollten sich die Webseite anschauen, um sicher zu gehen, dass sie nicht eine Lösung einer Übungsaufgabe oder einer Problemstellung enthält, die Sie stellen wollen. Wir erwarten, dass sich die Menge der veröffentlichten Lösungen kontinuierlich über die Zeit vergrößern wird, sodass Sie sich die Webseite jedesmal anschauen sollten, wenn Sie die Lehrveranstaltung anbieten.

Wir haben diejenigen Abschnitte und Übungen mit einem Stern (*) versehen, die eher für fortgeschrittene Studierende als für Studienanfänger geeignet sind. Ein mit Stern versehener Abschnitt ist nicht notwendigerweise schwieriger als ein Abschnitt ohne Stern, er kann aber mehr Verständnis von höherer Mathematik erfordern. Ebenso kann eine mit Stern versehene Übung ein höheres Niveau oder mehr als nur durchschnittliche Kreativität voraussetzen.

An die Studierenden

Wir hoffen, dass Ihnen dieses Lehrbuch eine unterhaltsame Einführung in das Gebiet der Algorithmen liefert. Wir haben uns bemüht, jeden Algorithmus leicht zugänglich und interessant zu gestalten. Um Ihnen zu helfen, wenn Sie auf ungewohnte oder schwierige Algorithmen stoßen, beschreiben wir jeden Algorithmus Schritt für Schritt. Wir liefern außerdem sorgfältige Erklärungen zur Mathematik, die notwendig ist, um die Analyse der Algorithmen zu verstehen. Wenn Sie bereits über einige Vorkenntnisse auf einem Gebiet verfügen, dann werden Sie feststellen, dass die Kapitel so eingerichtet sind, dass Sie die einführenden Abschnitte überfliegen und schnell mit dem höheren Stoff fortfahren können.

Dies ist ein umfangreiches Buch und Ihre Vorlesung wird wahrscheinlich nur einen Teil des enthaltenen Stoffes behandeln. Wir haben uns bemüht, dieses Buch so zu gestalten, dass es für Sie jetzt als Lehrbuch und später während Ihrer Karriere auch als

mathematisches Nachschlagewerk oder als technisches Handbuch nützlich sein wird. Was sind die fachlichen Voraussetzungen für die Lektüre dieses Buches?

- Sie sollten etwas Programmiererfahrung haben. Insbesondere sollten Sie rekursive Prozeduren und einfache Datenstrukturen, wie zum Beispiel Felder und verkettete Listen, verstehen.
- Sie sollten Übung mit mathematischen Beweisen, im besonderen mit mathematischer Induktion haben. Einige Teile dieses Buches stützen sich auf Kenntnisse in elementarer Analysis. Darüber hinaus vermitteln Ihnen die Teile I und VIII dieses Buches alle mathematischen Methoden, die Sie benötigen werden.

Wir haben laut und deutlich Ihre Forderung gehört, Lösungen zu Problemstellungen und Übungen bereitzustellen. Unsere Webseite <http://mitpress.mit.edu/algorithms> verlinkt zu Lösungen für einige der Problemstellungen und Übungen. Überprüfen Sie Ihre Lösungen mit unseren. Wir bitten aber darum, uns Ihre Lösungen nicht zuzuschicken.

An die Fachleute

Die große Auswahl an Themen in diesem Buch macht dieses zu einem ausgezeichneten Handbuch zum Thema Algorithmen. Da jedes Kapitel in sich relativ geschlossen ist, können Sie sich auf die Themen konzentrieren, die Sie am meisten interessieren.

Die meisten der hier besprochenen Algorithmen haben großen praktischen Nutzen. Daher befassen wir uns auch mit den Belangen der Implementierung und anderen technischen Fragen. Wir stellen in der Regel praktische Alternativen zu den wenigen Algorithmen vor, die vorrangig von theoretischem Interesse sind.

Falls Sie einen der Algorithmen implementieren möchten, dann sollte es für Sie ziemlich einfach sein, den Pseudocode in Ihre bevorzugte Programmiersprache zu übersetzen. Wir haben den Pseudocode so entworfen, dass er jeden Algorithmus klar und knapp präsentiert. Folglich befassen wir uns nicht mit Fehlerbehandlung und anderen softwaretechnischen Fragen, die spezielle Annahmen über Ihre Programmierumgebung erfordern. Wir versuchen, jeden Algorithmus einfach und genau darzustellen, ohne es den Eigenheiten einer speziellen Programmiersprache zu ermöglichen, dessen Wesen zu verdecken.

Wir sehen ein, dass Sie Ihre Lösungen zu Problemen und Übungen nicht mit den von einem Dozenten zur Verfügung gestellten Lösungen vergleichen können, wenn Sie dieses Buch im Selbststudium benutzen. Unsere Webseite <http://mitpress.mit.edu/algorithms> verlinkt zu Lösungen für einige der Problemstellungen und Übungen, sodass Sie Ihre Lösungen überprüfen können. Senden Sie uns aber bitte nicht Ihre Lösungen zu.

An unsere Kollegen

Wir haben ein umfangreiches Quellenverzeichnis und Hinweise auf die aktuelle Literatur bereitgestellt. Jedes Kapitel endet mit einer Reihe von Kapitelbemerkungen, die

historische Details und Hinweise geben. Die Kapitelbemerkungen bilden dennoch keine vollständige Referenz zum gesamten Gebiet der Algorithmen. Obwohl es von einem Buch dieses Umfangs schwer zu glauben sein mag, hinderte Platzmangel uns daran, viele interessante Algorithmen mit aufzunehmen.

Änderungen innerhalb der dritten Auflage

Was hat sich in der dritten Auflage in Bezug auf die zweite geändert? Die Anzahl der Änderungen entspricht in etwa der bei der zweiten Auflage. Wie wir bereits in Bezug auf diese Änderungen gesagt haben, hat sich das Buch also kaum oder ziemlich viel geändert.

Ein kurzer Blick in das Inhaltsverzeichnis zeigt, dass die meisten Kapitel und Abschnitte der zweiten Auflage auch in der dritten Auflage vorkommen. Wir haben zwei Kapitel und einen Abschnitt entfernt, haben dafür drei neue Kapitel und ansonsten zwei neue Abschnitte hinzugefügt.

Wir haben die gemischte Organisation der ersten beiden Auflagen beibehalten. Anstatt die Kapitel ausschließlich nach Problembereichen oder ausschließlich nach Techniken zu gliedern, findet man in diesem Buch beides vor. Es enthält technikbasierte Kapitel wie die Kapitel über die Teile-und-Beherrsche-Methode, dynamische Programmierung, Greedy-Algorithmen, amortisierte Analyse, NP-Vollständigkeit und Approximationsalgorithmen. Es enthält jedoch auch jeweils ganze Teile über Sortieren, Datenstrukturen für dynamische Mengen und Graphalgorithmen. Dies ist dadurch begründet, dass, wenn gleich Sie wissen müssen, wie Techniken zum Entwurf oder zur Analyse von Algorithmen anzuwenden sind, aus den Problemstellungen nicht unmittelbar für Sie zu ersehen ist, welche Technik am ehesten hilft, das Problem zu lösen.

Wir geben eine Zusammenfassung der wichtigsten Änderungen in der dritten Auflage:

- Wir haben neue Kapitel zu van-Emde-Boas-Bäume und mehrfädigen (engl.: *multithreaded*) Algorithmen eingefügt. Die Grundlagen von Matrizen haben wir in den Anhang verschoben.
- Wir haben das Kapitel zu Rekursionsgleichungen überarbeitet, sodass es nunmehr die Teile-und-Beherrsche-Methode besser abdeckt. Die ersten zwei Abschnitte wenden die Teile-und-Beherrsche-Methode an, um zwei verschiedene Problemstellungen zu lösen. Der zweite Abschnitt dieses Kapitels stellt Strassens Algorithmus zur Matrixmultiplikation vor, den wir aus dem Kapitel über Matrixoperationen hierhin verschoben haben.
- Wir haben zwei Kapitel, die nur selten gelehrt werden, entfernt: binomiale Heaps und Sortiernetzwerke. Eine zentrale Idee bei Sortiernetzwerken, das 0-1-Prinzip, findet sich in dieser Auflage als 0-1-Sortierlemma für Vergleiche-Vertausche Algorithmen in der Problemstellung 8-7 wieder. Die Behandlung von Fibonacci-Heaps beruht nicht mehr auf binomialen Heaps.
- Wir haben unsere Betrachtungen zu dynamischer Programmierung und Greedy-Algorithmen überarbeitet. Dynamische Programmierung leiten wir nun ab mit

einem interessanteren Problem, dem Schneiden von Stahlstangen, als mit der Ablaufkoordinierung von Montagebändern aus der zweiten Auflage. Zudem heben wir ein bisschen mehr auf Memoisation ab, als wir dies in der zweiten Auflage gemacht haben, und führen den Begriff des Teilproblem-Graphen als eine Möglichkeit, die Laufzeit eines auf dynamischer Programmierung beruhender Algorithmus zu verstehen, ein. In unserem Anfangsbeispiel zu Greedy-Algorithmen, dem Aktivitäten-Auswahl-Problem, leiten wir Greedy-Algorithmen direkter ab, als wir dies in der zweiten Auflage getan haben.

- Die Methode, mit der wir einen Knoten aus einem binären Suchbaum (insbesondere aus Rot-Schwarz-Bäumen) löschen, gewährleistet nun, dass genau der Knoten, der gelöscht werden soll, auch tatsächlich gelöscht wird. In den ersten beiden Auflagen konnte in bestimmten Fällen ein anderer Knoten gelöscht werden, wobei zuvor dessen Inhalt in den Knoten kopiert wurde, der der Lösche-Prozedur an sich übergeben worden war. Mit dieser neuen Methode, Knoten zu löschen, kann es nicht mehr passieren, dass andere Programmteile, die Zeiger auf Knoten des Baumes verwalten, irrtümlicherweise alte Zeiger auf Knoten, die gelöscht wurden, haben.
- Die Ausführungen zu Flussnetzwerken basieren nun vollständig auf Flüssen auf Kanten. Dieser Ansatz ist intuitiver als der über den Nettofluss aus den ersten beiden Auflagen.
- Da wir die Ausführungen zu den Grundlagen von Matrizen und zu Strassens Algorithmus in andere Kapitel verschoben haben, ist das Kapitel zu Matrixoperationen kürzer als in der zweiten Auflage.
- Wir haben die Behandlung des Knuth-Morris-Patt String-Matching Algorithmus abgeändert.
- Wir haben mehrere Fehler korrigiert. Die meisten von ihnen, wenn auch nicht alle, wurden uns über das auf unserer Webseite verfügbare Fehlerverzeichnis zugeschickt.
- Aufgrund vieler Anfragen haben wir die (bisherige) Syntax unseres Pseudocodes geändert. Wir benutzen nun, wie dies auch in C, C++, Java und Python der Fall ist, “=” für eine Zuweisung und “==” für den Test auf Gleichheit. Desgleichen haben wir die Schlüsselwörter **do** und **then** entfernt und übernehmen “//” als unser Kommentarsymbol, wenn der Rest der entsprechende Zeile Kommentar sein soll. Zudem benutzen wir Punktnotation für die Angabe von Objektattributen. Unser Pseudocode bleibt prozedural. Anders formuliert, anstatt Methoden auf Objekten laufen zu lassen, rufen wir einfach Prozeduren auf, denen wir Objekte als Parameter mitgeben.
- Wir haben 100 neue Übungsaufgaben und 28 neue Problemstellungen eingefügt. Wir haben auch viele Bibliographieinträge aktualisiert und mehrere neue hinzugefügt.

- Schlussendlich sind wir das ganze Buch durchgegangen und haben Sätze, Absätze und Abschnitte umgeschrieben, damit die Ausführungen klarer werden und der Schreibstil lebhafter wird.

Internetpräsenz

Sie können unsere Webseite <http://mitpress.mit.edu/algorithms/> nutzen, um zusätzliche Informationen zu bekommen und um mit uns zu kommunizieren. Die Webseite verlinkt zu einer Liste bekannter Fehler, zu Lösungen ausgewählter Übungsaufgaben und Problemen sowie zu anderem Inhalt, den wir gegebenenfalls noch bereitstellen werden. Zudem erklärt sie (natürlich) die blöden Professorenwitze. Die Webseite sagt Ihnen auch, wie Sie uns Fehler oder Anregungen zukommen lassen können.

Wie wir das Buch hergestellt haben²

Wie die zweite Auflage haben wir auch die dritte Auflage mit $\text{\LaTeX} 2_{\epsilon}$ erstellt. Wir benutzten den Times Font mit dem auf den MathTime Pro 2 Fonts basierenden mathematischen Schriftsatz. Wir bedanken uns für die entsprechende technische Unterstützung bei Michael Spivak von Publish or Perish, Inc., Lance Carnes von Personal TeX, Inc., und Tim Tregubov vom Dartmouth College. Wie in den vorangehenden zwei Auflagen haben wir den Index mit Windex, einem C Programm, das wir geschrieben haben, übersetzt. Das Literaturverzeichnis wurde mit $\text{BIB}\text{\TeX}$ erzeugt. Die PDF-Dateien für dieses Buch wurden auf einem MacBook unter OS 10.5 generiert.

Die Abbildungen der dritten Auflage haben wir mit MacDraw Pro gezeichnet, unter Zuhilfenahme des psfrag Pakets von $\text{\LaTeX} 2_{\epsilon}$ zur Darstellung mathematischer Ausdrücke in den Abbildungen. Leider ist MacDraw Pro eine alte Software, die seit über einem Jahrzehnt nicht mehr verkauft wird. Glücklicherweise haben wir noch eine Handvoll Macintoshes, die unter dem Betriebssystem OS 10.4 laufen und so MacDraw Pro ausführen können – jedenfalls meistens. Sogar unter dieser alten Umgebung finden wir MacDraw Pro um ein Vielfaches einfacher zu benutzen als irgendeine andere Zeichensoftware, wenn es um Zeichnungen geht, die Informatiktexte begleiten sollen. Zudem erzeugt dieses Programm schöne Ausgaben.³ Wer weiß, wie lange unsere alten Macs noch laufen werden; also falls jemand von Apple zuhört: *Bitte bauen Sie eine MacDraw Pro Version, die unter OS X läuft!*

²Diese Ausführungen beziehen sich zum Teil nur auf die dritte Auflage der *englischen* Ausgabe.

³Wir haben uns mehrere Zeichenprogramme, die unter Mac OS X laufen, angeschaut, aber alle haben erhebliche Nachteile verglichen mit MacDraw Pro. Wir haben kurz versucht, die Abbildungen für dieses Buch mit einem anderen, gut bekannten Zeichenprogramm zu erstellen. Es stellte sich heraus, dass es wenigstens fünf Mal so lange dauert, eine Abbildung zu erzeugen, als mit MacDraw Pro, und das Ergebnis sah schlechter aus. Aus diesem Grund unsere Entscheidung, zu den alten Macintoshes und MacDraw Pro zurückzukehren.

Danksagungen zur dritten Auflage

Wir arbeiten nunmehr seit über zwei Jahrzehnte mit The MIT Press zusammen und was war das für eine wunderbare Zusammenarbeit! Wir danken Ellen Faran, Bob Prior, Ada Brunstein und Mary Reilly für ihre Hilfe und Unterstützung.

Während der Erstellung der dritten Auflage arbeiteten wir an geografisch unterschiedlichen Orten, am Dartmouth College Department of Computer Science, im MIT Computer Science and Artificial Intelligence Laboratory, und am Columbia University Department of Industrial Engineering and Operations Research. Wir danken unseren jeweiligen Universitäten und Kollegen, dass wir in dermaßen unterstützenden und anregenden Umgebungen arbeiten konnten.

Julie Sussman, P.P.A., stand uns erneut als Lektorin zur Verfügung. Wieder und wieder waren wir verblüfft über die Fehler, die wir übersehen hatten, die Julie aber fand. Sie half uns auch, die Darstellung an mehreren Stellen zu verbessern. Gäbe es eine Ruhmeshalle für Lektorinnen und Lektoren, Julie wäre ein Platz sicher. Sie ist einfach nur phänomenal. Danke, danke, danke, Julie! Priya Natarajan fand ebenfalls einige Fehler, die wir noch vor Drucklegung korrigieren konnten. Alle Fehler, die sich im Buch noch befinden (und es werden sicherlich welche geben) liegen in der Verantwortung der Autoren (und wurden vermutlich eingefügt, nachdem Julie den Text gelesen hat).

Die Behandlung der van-Emde-Boas-Bäume basiert auf Notizen von Erik Demaine, die wiederum durch Michael Bender beeinflusst sind. Wir haben auch Ideen von Javed Aslam, Bradley Kuszmaul und Hui Zha in diese Auflage eingearbeitet.

Das Kapitel über Multithreading basiert auf Notizen, die ursprünglich zusammen mit Harald Prokop geschrieben worden sind, und wurde beeinflusst durch mehrere andere Kollegen, die am Cilk Projekt des MIT beteiligt sind. Zu nennen sind insbesondere Bradley Kuszmaul und Matteo Frigo. Der Entwurf des mehrfädigen Pseudocodes wurde durch die MIT Cilk Erweiterung von C und die Cilk Arts Cilk++ Erweiterung von C++ inspiriert.

Wir bedanken uns auch bei den vielen Lesern der ersten und zweiten Auflage, die uns auf Fehler aufmerksam machten oder uns Anregungen zukommen ließen, wie dieses Buch verbessert werden könnte. Wir korrigierten alle echten Fehler, die uns gemeldet worden sind, und haben so viele Anregungen, wie wir konnten, eingebunden. Wir sind glücklich darüber, dass die Anzahl dieser Mitwirkenden so groß geworden ist, dass wir bedauern müssen, dass es nicht mehr möglich ist, alle aufzuzählen.