
Springer-Lehrbuch

Stefan Hougardy · Jens Vygen

Algorithmische Mathematik

 Springer Spektrum

Stefan Hougardy
Jens Vygen
Forschungsinstitut für Diskrete Mathematik
Universität Bonn
Bonn, Deutschland

ISSN 0937-7433

ISBN 978-3-662-47013-8

ISBN 978-3-662-47014-5 (eBook)

DOI 10.1007/978-3-662-47014-5

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

Mathematics Subject Classification (2010): 00-01, 00A69, 68-01, 65-01, 68Q01, 68R01, 68W01, 68N15, 05C85, 90C27, 65F05

Springer Spektrum

© Springer-Verlag Berlin Heidelberg 2016

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, 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.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen.

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier.

Springer-Verlag GmbH Berlin Heidelberg ist Teil der Fachverlagsgruppe Springer Science+Business Media
(www.springer.com)

Vorwort

Seit es Computer gibt, nimmt die Bedeutung von Algorithmen in fast allen Bereichen der Mathematik ständig zu. An der Universität Bonn wurde daher neben Analysis und Linearer Algebra eine dritte Grundvorlesung für das erste Semester konzipiert: die Algorithmische Mathematik. Dieses Buch gibt genau die Inhalte dieser Vorlesung wieder, die die Autoren mehrfach gehalten haben, und die etwa 30 mal 90 Minuten (zuzüglich Übungen) umfasst. Wir setzen nirgends mehr als Schulwissen voraus; dennoch ist das Buch für Leser ohne mathematische Vorbildung anspruchsvoll.

Im Gegensatz zu den meisten anderen einführenden Büchern über Algorithmen, die vielleicht eher auf Informatikstudenten abzielen, legen wir von Anfang an viel Wert auf eine rigorose mathematische Vorgehensweise. Exakte Definitionen, präzise Sätze und genau ausgearbeitete elegante Beweise sind gerade am Anfang eines Mathematikstudiums unentbehrlich. Das Buch beinhaltet aber auch viele Beispiele, Erläuterungen und Hinweise auf weiterführende Themen.

Bei der Auswahl der Themen haben wir darauf geachtet, ein möglichst breites Spektrum von Algorithmen und algorithmischen Fragestellungen zu zeigen, soweit dies ohne tiefere mathematische Kenntnisse möglich ist. Wir behandeln Grundlagen (Kap. 1–3), numerische Fragen (Kap. 4–5), Graphen (Kap. 6–7), Sortieralgorithmen (Kap. 8), Kombinatorische Optimierung (Kap. 9 und 10) sowie die Gauß-Elimination (Kap. 11). Dabei sind die verschiedenen Themen oft miteinander verzahnt; die Reihenfolge kann daher nicht ohne Weiteres verändert werden. Neben klassischen Algorithmen und deren Analyse wird der Leser wichtige theoretische Grundlagen, viele Querverbindungen und sogar auch Hinweise auf offene Forschungsfragen entdecken.

Algorithmen wirklich zu verstehen und mit ihnen zu arbeiten ist kaum möglich, ohne sie auch implementieren zu können. Parallel zu den mathematischen Themen führen wir daher in diesem Buch in die Programmiersprache C++ ein. Wir bemühen uns dabei, die technischen Details auf das Notwendige zu beschränken — dies ist kein Programmierkurs! — und dennoch das Buch auch für Studienanfänger ohne Programmiererfahrung zugänglich zu machen.

Die von uns sorgfältig konzipierten Programmbeispiele sollen einerseits die wichtigsten Elemente der Sprache C++ lehren und darüber hinaus zum Selbststudium anregen. Andererseits sind sie aber auch stets so gewählt, dass sie thematisch den jeweiligen Stoff

ergänzen. Natürlich kann man nicht wirklich programmieren lernen, ohne es selbst zu tun, ebenso wenig wie man Mathematik lernen kann, ohne selbst Aufgaben und Probleme zu lösen. Dazu möchten wir alle Studienanfänger von Beginn an mit Nachdruck ermuntern.

Wir wünschen allen Lesern viel Freude an der Algorithmischen Mathematik!

Bonn, März 2015

Stefan Hougardy und Jens Vygen

Anmerkungen zu den C++-Programmen

Dieses Buch enthält eine Reihe von Programmbeispielen in C++. Der Sourcecode aller dieser Programme kann über die Webseiten der Autoren heruntergeladen werden. Wir benutzen in diesem Buch die in ISO/IEC 14882:2011 [6] spezifizierte C++-Version, die auch unter dem Namen C++11 bekannt ist. Zum Kompilieren der Programmbeispiele eignen sich alle gängigen C++-Compiler, die diese C++-Version unterstützen. Beispielsweise unterstützt der frei verfügbare GNU C++ Compiler g++ ab der Version 4.8.1 alle in diesem Buch benutzten Sprachelemente von C++11. Gute Lehrbücher zu C++11 sind z. B. [5, 26, 33]. Ausführliche Informationen zu C++11 findet man auch im Internet, z. B. unter <http://de.cppreference.com> oder <http://www.cplusplus.com>.

Danksagung

Wir möchten uns an dieser Stelle bei allen bedanken, die uns im Laufe der Jahre Anregungen und Verbesserungsvorschläge zu diesem Buch gegeben haben. Neben den Studierenden aus unseren Vorlesungen möchten wir uns hier insbesondere bei Christoph Bartoschek, Ulrich Brenner, Helmut Harbrecht, Stephan Held, Dirk Müller, Philipp Ochsendorf, Jan Schneider und Jannik Silvanus bedanken.

Für Hinweise auf verbleibende Fehler und weitere Verbesserungsvorschläge sind wir natürlich jederzeit dankbar.

Stefan Hougardy und Jens Vygen

Symbolverzeichnis

	mit der Eigenschaft, dass
\exists	es gibt (mindestens) ein
\forall	für alle
\emptyset	leere Menge
\subseteq	Teilmenge
\subset	echte Teilmenge
\cup	Vereinigung von Mengen
\cap	Schnitt von Mengen
$\dot{\cup}$	disjunkte Vereinigung von Mengen
Δ	symmetrische Differenz
\times	kartesisches Produkt
\wedge	logisches und
\vee	logisches oder
[.]	obere Gaußklammer
[.]	untere Gaußklammer
\approx	ungefähr gleich
\leftarrow	Zuweisung im Pseudocode
\top	Transposition

Inhaltsverzeichnis

1	Einleitung	1
1.1	Algorithmen	1
1.2	Berechnungsprobleme	2
1.3	Algorithmen, Pseudocode und C++	4
1.4	Einfacher Primzahltest	6
1.5	Sieb des Eratosthenes	12
1.6	Nicht alles ist berechenbar	15
2	Darstellungen ganzer Zahlen	21
2.1	b -adische Darstellung natürlicher Zahlen	21
2.2	Exkurs: Aufbau des Hauptspeichers	25
2.3	b -Komplementdarstellung ganzer Zahlen	27
2.4	Rationale Zahlen	30
2.5	Beliebig große ganze Zahlen	35
3	Rechnen mit ganzen Zahlen	41
3.1	Addition und Subtraktion	41
3.2	Multiplikation	42
3.3	Euklidischer Algorithmus	44
4	Approximative Darstellungen reeller Zahlen	49
4.1	b -adische Darstellung reeller Zahlen	49
4.2	Maschinenzahlen	51
4.3	Rundung	53
4.4	Maschinenzahlenarithmetik	55

5	Rechnen mit Fehlern	57
	5.1 Binäre Suche	58
	5.2 Fehlerfortpflanzung	59
	5.3 Kondition	61
	5.4 Fehleranalyse	62
	5.5 Newton-Verfahren	63
6	Graphen	67
	6.1 Grundlegende Definitionen	67
	6.2 Wege und Kreise	69
	6.3 Zusammenhang und Bäume	71
	6.4 Starker Zusammenhang und Arboreszenzen	73
	6.5 Exkurs: Elementare Datenstrukturen	75
	6.6 Darstellungen von Graphen	78
7	Einfache Graphenalgorithmen	85
	7.1 Graphendurchmusterung	85
	7.2 Breitensuche	87
	7.3 Bipartite Graphen	89
	7.4 Azyklische Digraphen	90
8	Sortieralgorithmen	93
	8.1 Das allgemeine Sortierproblem	93
	8.2 Sortieren durch sukzessive Auswahl	94
	8.3 Sortieren nach Schlüsseln	99
	8.4 Mergesort	100
	8.5 Quicksort	102
	8.6 Binäre Heaps und Heapsort	104
	8.7 Weitere Datenstrukturen	109
9	Optimale Bäume und Wege	111
	9.1 Optimale aufspannende Bäume	111
	9.2 Implementierung von Prim's Algorithmus	114
	9.3 Kürzeste Wege: Dijkstra's Algorithmus	117
	9.4 Konservative Kantengewichte	120
	9.5 Kürzeste Wege mit beliebigen Kantengewichten	122
10	Matching und Netzwerkflüsse	125
	10.1 Das Matching-Problem	125
	10.2 Bipartites Matching	126
	10.3 Max-Flow-Min-Cut-Theorem	128
	10.4 Algorithmen für maximale Flüsse	131

11	Gauß-Elimination	135
	11.1 Die Operationen der Gauß-Elimination	137
	11.2 LU-Zerlegung	140
	11.3 Gauß-Elimination mit rationalen Zahlen	143
	11.4 Gauß-Elimination mit Maschinenzahlen	146
	11.5 Matrixnormen	149
	11.6 Kondition linearer Gleichungssysteme	151
	Literatur	157
	Sachverzeichnis	159