

Günther Stiege  
Einführung in die Informatik

Januar 2013



Berichte aus der Informatik

**Günther Stiege**

**Einführung in die Informatik**

Shaker Verlag  
Aachen 2013

### **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.

Prof. Dr. Günther Stiege  
Carl von Ossietzky Universität Oldenburg  
Department für Informatik

Copyright Shaker Verlag 2013

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8440-1661-1  
ISSN 0945-0807

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen  
Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9  
Internet: [www.shaker.de](http://www.shaker.de) • E-Mail: [info@shaker.de](mailto:info@shaker.de)

## *Vorwort*

*Sie wartet im Park bei den Tannen.  
 Er hat noch viel zu tun:  
 Den Back gepätscht.  
 Pettern gemetscht.  
 An der Platine den Finger gequetscht.  
 Igemeht.  
 Einen Apfel geschält.  
 Die Datei abgedätet.  
 Er hat sich arg verspätet.  
 Nun  
 geht sie verärgert von dannen.*

*chg escalera 1999*

Das vorliegende Buch ist aus einem Skript zu den Vorlesungen „Programmierung“ und „Datenstrukturen“ entstanden, die ich im Wintersemester 1998/99 und im Sommersemester 1999 an der Universität Oldenburg gehalten habe. Diese beiden Vorlesungen bildeten einen einführenden, in erster Linie für Anfänger im Diplomstudiengang Informatik bestimmten Kurs. Den Diplomstudiengang Informatik gibt es nicht mehr. Eine anspruchsvolle einführende Darstellung der Informatik ist jedoch auch für den universitären Bachelor-Studiengang in Informatik notwendig. Sie wird auch in naturwissenschaftlichen und technischen universitären Studiengängen gebraucht. Für diesen Leserkreis ist das Buch gedacht. Ich habe den Stoff in die Teile „Grundlagen“, „Algorithmen“, „Einfache Datenstrukturen“, „Allgemeine Graphen“ und „Parallelität“ eingeteilt. Das soll kurz erläutert werden.

In den Grundlagen wird mit einführenden Beispielen im 1. Kapitel ein Einstieg in den Stoff gefunden. Ich habe es für zweckmäßig gehalten, Knuth [Knut1997] sowie einer Vorlesung meines Kollegen R. Vollmar zu folgen und mit dem guten, alten euklidischen Algorithmus zu beginnen. Die dazugehörigen Effizienzbetrachtungen sind zugegebenermaßen für Anfänger in den ersten Vorlesungstunden nicht ganz einfach. Hier wie auch im Rest des Buches habe ich jedoch Wert darauf gelegt, mathematischen Schwierigkeiten, insbesondere auch Beweisen, nicht auszuweichen. Die leider in der Informatikausbildung zunehmend zu beobachtende Tendenz, Schulung in formalem Denken zugunsten praktischer Fertigkeiten und Kenntnisse zu reduzieren, halte ich für fatal.

Das ausführliche 2. Kapitel erläutert anhand der Sprache C die Grundlagen der Pro-

grammierung. Auch hierzu einige Anmerkungen. Es ist üblich, in der universitären Informatikausbildung außer dem Informatik-Grundkurs einen getrennten Programmierkurs vorzusehen. Meistens wird dort eine „moderne“ Sprache, objektorientiert oder funktional, zu Grunde gelegt. Ich bin bewußt davon abgewichen, da ich eine explizite und intensive Beschäftigung mit Konzepten wie Zeigern und Unterprogrammen in der Informatikausbildung für sehr wichtig und prägend halte.

Die Kapitel 3 „Darstellung von Daten durch Bitmuster“ und 4 „Rechensysteme“ vermitteln so viel über „reale“ Rechner, wie für eine Einführung nötig ist.

Der Kern des Kurses sind die Teile II, III und IV. In Teil II werden in zwei Kapiteln Algorithmen allgemein untersucht. Es wird der naive Algorithmus-Begriff behandelt und es werden Markov-Methoden als Beispiel für eine Formalisierung des Algorithmus-Begriffs vorgestellt. Die Churchsche These wird erwähnt und die Unlösbarkeit des Halteproblems gezeigt. Es wird die Zeiteffizienz von Algorithmen betrachtet und dazu die O-Notation eingeführt. Zum Schluß wird das Problem  $\mathcal{P} = \mathcal{NP}$  erläutert. In zwei ergänzenden Abschnitten wird in Form eines kurzen Überblicks auf Turingmaschinen und Formale Sprachen und auf die näherungsweise Lösung schwieriger Probleme eingegangen.

Teil III ist einfachen Datenstrukturen, nämlich Listen, Suchbäumen, Hashing und Sortieren gewidmet. Die entsprechenden Datenstrukturen und Algorithmen werden vorgestellt und untersucht.

Teil IV des Buches handelt von allgemeinen Graphen. Das sind Graphen, in denen ungerichtete Kanten und gerichtete Bögen beliebig gemischt auftreten. Diese vereinheitlichende Sicht ist neu und war mir zu der Zeit, in der der Kurs durchgeführt wurde, noch nicht bekannt. Aus diesem Grund ist Teil IV des Buches ganz anders, insbesondere auch umfangreicher, als die graphentheoretischen Kapitel des ursprünglichen Skripts. Den Kern des vierten Teils bilden die Kapitel „Grundlagen“, „Darstellungen“, „Wege und Zusammenhang“, „Tiefensuche und Breitensuche“ sowie „Die Biblockzerlegung“.

In der Informatik und anderen Anwendungsgebieten der Graphentheorie sind Netzwerke, also Graphen, deren Knoten und/oder Linien zusätzliche Attribute aufweisen, von besonderer Bedeutung. Als wichtiges Beispiel werden „Kürzeste Wegen“ in allgemeinen Graphen untersucht.

Parallelität ist ein sehr wichtiges Gebiet der Informatik. In Einführungen und Grundkursen bleibt oft keine Zeit, es zu behandeln. Einen ersten Einblick geben die zwei Kapitel von Teil V. Das Kapitel „Parallelität in Rechensystemen und Netzen“ beschreibt, wo und in welcher Form parallele Abläufe auftreten. Das Kapitel „Programmieren II: Parallele Programme“ behandelt am Beispiel von Leichtgewichtsprozessen in C einige einführende Fragen der Parallelität.

Ich habe mich bemüht, die nach meiner Meinung wesentlichen Dinge eines jeden angesprochenen Gebietes hinreichend ausführlich darzustellen. Eine Reihe von Dingen, die ich auch für wichtig halte, die man bei Zeitmangel aber überspringen kann, habe ich explizit mit

einem \* als Ergänzung gekennzeichnet. Es sind einzelne Abschnitte, in der Graphentheorie auch ganze Kapitel. Einige Ergänzungen haben nur Überblickscharakter. In Ergänzungen wird durchaus auf andere Ergänzungen zurückgegriffen, an anderer Stelle nicht.

Zu den meisten Kapiteln gibt es Übungen. Zur Mehrzahl der Übungen sind im Anhang Lösungen oder Lösungshinweise angegeben. Die Übungen reichen für einen ergänzenden Übungskurs nicht aus, denn reine Lernaufgaben zum Trainieren des Stoffes habe ich nicht aufgenommen. Statt dessen habe ich für die Übungen Erweiterungen und Ergänzungen ausgewählt. An mehr als einer Stelle wird der Leser auch aufgefordert, einfache Aussagen des Stoffteils selber zu beweisen.

In den Anhängen zum Buch befinden sich eine Zusammenfassung mathematischer Hilfsmittel allgemeiner Art, eine kurze Darstellung der benötigten Begriffe und Ergebnisse aus der Mengenlehre, ein Anhang über wichtige Grundlagen der Wahrscheinlichkeitstheorie und einiges mehr. Vielleicht helfen sie ein wenig, über die Schwierigkeiten, die Anfänger mit der Sprech- und Denkweise der Hochschulmathematik haben, hinwegzukommen. Zu diesem Zweck sei auch das Buch *Mathe-Toolbox* von Schöning und Kestler [SchoK2012] empfohlen.

Für die Vorlesung, das Skript und jetzt das Buch habe ich eine Vielzahl von Lehrbüchern und andere schriftliche Quellen benutzt. Sie sind als Literaturzitate im Text der einzelnen Kapitel, speziell in den abschließenden Abschnitten „Literatur“ angegeben. Kurs und Skript wurden in besonderem Maße von den Büchern von Aho/Ullman [AhoU1995], Cormen/Leiserson/Rivest [CormLR1990], Knuth ([Knut1997], [Knut1998], [Knut1998a]) sowie Kowalk [Kowa1996] beeinflusst.

Eine Reihe von Verzeichnissen ergänzt das Buch. Besonderen Wert habe ich auf ein ausführliches Stichwortverzeichnis gelegt. Insgesamt ergibt sich ein ziemlich dickes Buch. Es würde mich freuen, wenn es als ganzes oder in Teilen zu intensivem Lesen anregte. Besonders erfreut wäre ich, wenn es Neulingen in der Informatik etwas von der Faszination und der Schönheit dieser Wissenschaft vermittelte.

Das Buch ist in  $\text{\LaTeX}$  geschrieben und nach amerikanischem Brauch wird ein “chapter” in “sections” und diese in “subsections” unterteilt. Nur widerstrebend habe ich mich durchgerungen, deshalb bei der Kapitelgliederung von „Abschnitten“ (statt Unterkapitel) und „Unterabschnitten“ zu sprechen.

Vielen Leuten gilt es zu danken:

Meine Mitarbeiter Björn Briel, Olaf Maibaum und Ingo Stierand haben mir bei der Erstellung des Skripts nicht nur durch intensives Korrekturlesen, sondern auch durch viele fachliche Diskussionen und Anregungen immer wieder geholfen. Auch Anregungen von Studierenden sind in die endgültige Fassung eingeflossen. Etliche Ungenauigkeiten und eine Vielzahl von Schreibfehlern konnten dadurch ausgemerzt werden. Auch Herr Michael Uelschen und Frau Susanne Steiner haben durch Korrekturlesen zur Verbesserung beigetragen. Meinem Kollegen Hermann Luttermann danke ich für die Durchsicht des Kapitels

über Parallelität in Rechensystemen und Netzen.

Der nach der Vorlesung und der ersten Skriptversion entstandene umfangreiche Teil über allgemeine Graphen, wäre ohne die mehrjährige intensive Zusammenarbeit mit Ingo Stierand und Sergej Alekseev nicht möglich gewesen. Leider sind die interessanten und wichtigen Ergebnisse zur Ablaufanalyse aus der Dissertation Alekseev [Alek2006] wegen Platzmangels nicht mehr in das Buch eingeflossen.

Natürlich braucht es für die Entstehung eines Buches wie diesem eine vorherige Reifezeit sowie eine anregende Arbeitsumgebung. Beides habe ich im Kreis der Kollegen und meiner Mitarbeiter an meinen Wirkungsstätten, der TU Braunschweig, der Universität Hildesheim und der Universität Oldenburg gehabt.

Ein besonderer Dank gilt meinem Sohn Harold, der mit viel Zähigkeit den für ihn schwierigen Text Korrektur gelesen hat. Und natürlich meiner Frau, ohne deren Unterstützung ich das Buch gar nicht hätte schreiben können.

Hannover, im November 2012

G. Stiege

# Inhaltsverzeichnis

<i>Vorwort</i>	i
<b>I Grundlagen</b>	<b>1</b>
<b>1 Einführende Beispiele</b>	<b>3</b>
1.1 Der euklidische Algorithmus . . . . .	3
1.1.1 Entwicklung und Untersuchung des Algorithmus . . . . .	3
1.1.2 Ein Programm für den euklidischen Algorithmus . . . . .	6
1.1.3 Effizienz des euklidischen Algorithmus . . . . .	10
1.2 Sortieren durch Einfügen . . . . .	18
1.2.1 Allgemeines zum Sortieren . . . . .	18
1.2.2 Insertion Sort: Algorithmus und Programm . . . . .	18
1.2.3 Effizienz von Sortieren durch Einfügen . . . . .	19
1.3 Ein Kochrezept . . . . .	25
Aufgaben . . . . .	27
Literatur . . . . .	28
<b>2 Programmieren I: Sequentielle Programme</b>	<b>29</b>
2.1 Schichtenaufbau eines Rechensystems . . . . .	29
2.2 Dateien und Datenbanken . . . . .	33
2.3 Grundbegriffe . . . . .	34
2.4 Werte, Variable, Zeiger, Namen . . . . .	37
2.5 Wertebereiche, Operationen, Ausdrücke . . . . .	45
2.5.1 Ganze Zahlen . . . . .	45
2.5.2 Rationale Zahlen . . . . .	48
2.5.3 Zeichen . . . . .	52
2.5.4 Wahrheitswerte . . . . .	54
2.5.5 Adressen . . . . .	57
2.5.6 Aufzählungstypen . . . . .	59
2.6 Reihungen, Zeichenreihen, Sätze . . . . .	61

2.6.1	Reihungen . . . . .	61
2.6.2	Zeigerarithmetik in C . . . . .	65
2.6.3	Zeichenreihen . . . . .	67
2.6.4	Sätze . . . . .	68
2.7	Programmsteuerung . . . . .	78
2.8	Unterprogramme . . . . .	83
	Literatur . . . . .	91
<b>3</b>	<b>Darstellung von Daten durch Bitmuster</b>	<b>93</b>
3.1	Bits und Bitmuster . . . . .	94
3.2	Darstellung natürlicher Zahlen . . . . .	96
3.3	Darstellung ganzer Zahlen . . . . .	101
3.4	Darstellung rationaler Zahlen . . . . .	105
3.5	Hexadezimaldarstellung von Bitmustern . . . . .	109
3.6	Darstellung von Zeichenreihen . . . . .	112
3.7	Eigentliche Bitmuster und Wahrheitswerte . . . . .	117
3.8	Bitmuster in C . . . . .	118
	Literatur . . . . .	119
<b>4</b>	<b>Rechensysteme</b>	<b>121</b>
4.1	Grobschema der konventionellen Maschine . . . . .	121
4.1.1	Prozessor . . . . .	121
4.1.2	Hauptspeicher . . . . .	122
4.1.3	Ein-/Ausgabeschnittstellen . . . . .	126
4.1.4	Übertragungsmedium . . . . .	127
4.1.5	Periphere Geräte . . . . .	127
4.2	Prozessor und Maschinenbefehle . . . . .	129
4.2.1	Register des Prozessors . . . . .	129
4.2.2	Maschinenbefehle und Unterbrechungen . . . . .	133
4.2.3	Befehlsklassen . . . . .	135
4.2.4	Adressierungsarten . . . . .	137
4.3	Grundsoftware . . . . .	138
4.3.1	Betriebssystem . . . . .	139
4.3.2	Weitere Programme der Grundsoftware . . . . .	144
4.4	Virtuelle Adressierung . . . . .	145
	Literatur . . . . .	147
<b>II</b>	<b>Algorithmen</b>	<b>149</b>
<b>5</b>	<b>Algorithmen I: Naiv und formalisiert</b>	<b>151</b>

5.1	Der naive Algorithmus-Begriff . . . . .	152
5.1.1	Eigenschaften . . . . .	152
5.1.2	Schrittweise Verfeinerung . . . . .	155
5.1.3	Entwurfstechniken für Algorithmen . . . . .	159
5.1.4	Algorithmen in applikativer Darstellung . . . . .	162
5.2	Der formalisierte Algorithmusbegriff . . . . .	166
5.2.1	Markov-Algorithmen . . . . .	166
5.2.2	Churchsche These . . . . .	169
5.2.3	Algorithmisch unlösbare Probleme . . . . .	170
5.2.4	Turing-Maschinen und formale Sprachen* . . . . .	172
	Aufgaben . . . . .	175
	Literatur . . . . .	175
<b>6</b>	<b>Algorithmen II: Effizienz und Komplexität</b>	<b>177</b>
6.1	Laufzeitanalyse von Algorithmen und Programmen . . . . .	177
6.1.1	Vorbemerkungen . . . . .	177
6.1.2	Beispiel: Mischsortieren . . . . .	178
6.1.3	Boden und Decke . . . . .	189
6.1.4	Modulo . . . . .	193
6.1.5	Abschätzungen und Größenordnungen . . . . .	195
6.1.6	Rekurrenzen und erzeugende Funktionen* . . . . .	199
6.2	Die Komplexität von Problemen . . . . .	199
6.2.1	Überblick . . . . .	199
6.2.2	Komplexität von Algorithmen . . . . .	200
6.2.3	Beispiel: Hamiltonkreise . . . . .	201
6.2.4	Die Problemklassen $\mathcal{P}$ , $\mathcal{NP}$ und weitere . . . . .	210
6.3	Näherungslösungen schwieriger Probleme* . . . . .	216
	Aufgaben . . . . .	223
	Literatur . . . . .	223
<b>III</b>	<b>Einfache Datenstrukturen</b>	<b>225</b>
<b>7</b>	<b>Allgemeines zu Datenstrukturen</b>	<b>227</b>
7.1	Sätze und Vertretersätze . . . . .	227
7.2	Schlüssel . . . . .	228
<b>8</b>	<b>Listen</b>	<b>231</b>
8.1	Terminologie und Grundlagen . . . . .	231
8.1.1	Definition und Beispiele . . . . .	231
8.1.2	Operationen mit Listen . . . . .	232

8.2	Binärsuche . . . . .	235
8.3	Realisierung von Listen . . . . .	239
8.3.1	Verkettete Listen . . . . .	239
8.3.2	Realisierung von Listen durch Reihungen . . . . .	243
8.4	Keller, Schlangen, Halden . . . . .	245
8.4.1	Keller . . . . .	245
8.4.2	Keller: Beispiele und Anwendungen . . . . .	247
8.4.3	Schlangen . . . . .	251
8.4.4	Halden und Prioritätswarteschlangen . . . . .	252
	Aufgaben . . . . .	255
	Literatur . . . . .	255
<b>9</b>	<b>Suchbäume</b>	<b>257</b>
9.1	Binäre Suchbäume . . . . .	257
9.1.1	Allgemeines zum Suchen . . . . .	257
9.1.2	Binärbäume . . . . .	258
9.1.3	Binäre Suchbäume und Operationen auf ihnen . . . . .	266
9.2	Rot-Schwarz-Bäume . . . . .	277
9.2.1	Definition und Eigenschaften von Rot-Schwarz-Bäumen . . . . .	277
9.2.2	Operationen auf Rot-Schwarz-Bäumen . . . . .	280
9.3	Zufällige binäre Suchbäume . . . . .	293
9.4	B-Bäume und externe Datenspeicherung . . . . .	296
9.4.1	Mehrweg-Suchbäume . . . . .	296
9.4.2	Speicherung bei Dateien und Datenbanken . . . . .	298
9.4.3	B-Bäume . . . . .	299
9.5	Digitale Bäume . . . . .	303
	Aufgaben . . . . .	303
	Literatur . . . . .	305
<b>10</b>	<b>Suchen mit Schlüsseltransformation</b>	<b>307</b>
10.1	Direkte Speicherung . . . . .	308
10.2	Grundlagen des Hashings . . . . .	308
10.2.1	Hashingalgorithmen . . . . .	309
10.2.2	Kollisionsauflösung durch Verkettung . . . . .	311
10.2.3	Kollisionsauflösung durch offenes Hashing . . . . .	312
10.2.4	Universelles Hashing . . . . .	316
10.2.5	Dynamisches Hashing . . . . .	316
	Aufgaben . . . . .	317
	Literatur . . . . .	317

<b>11 Sortieren</b>	<b>319</b>
11.1 Allgemeines zum Sortieren . . . . .	319
11.2 Quicksort . . . . .	320
11.2.1 Algorithmus und Programm . . . . .	320
11.2.2 Komplexität von Quicksort . . . . .	326
11.2.3 Randomisiertes Quicksort . . . . .	328
11.2.4 Quicksort und Mischsortieren . . . . .	329
11.3 Halden, Heapsort und Prioritätswarteschlangen . . . . .	329
11.3.1 Halden (als Datenstruktur) . . . . .	329
11.3.2 Sortieren mit Halden (Heapsort) . . . . .	332
11.3.3 Prioritätswarteschlangen (nach Knuth) . . . . .	333
11.4 Mindestkomplexität beim Sortieren . . . . .	337
11.5 Lineares Sortieren . . . . .	339
Aufgaben . . . . .	342
Literatur . . . . .	343
<b>IV Allgemeine Graphen</b>	<b>345</b>
<b>12 Grundlagen allgemeiner Graphen</b>	<b>347</b>
12.1 Definitionen und Beispiele . . . . .	347
12.2 Orientierungsklassen und Untergraphen . . . . .	351
12.3 Bipartite Graphen . . . . .	353
12.4 Der Grad eines Knotens . . . . .	355
12.5 Anmerkung: Ersetzung von Kanten durch Bögen . . . . .	357
12.6 Gleichheit und Isomorphie von Graphen* . . . . .	357
Aufgaben . . . . .	359
Literatur . . . . .	360
<b>13 Darstellungen von Graphen</b>	<b>361</b>
13.1 Graphische Darstellung . . . . .	361
13.2 Darstellung durch Matrizen . . . . .	362
13.3 Darstellung durch Listen . . . . .	365
13.4 Externe Darstellungen und Basiswerkzeuge . . . . .	369
Aufgaben . . . . .	371
Literatur . . . . .	371
<b>14 Wege und Zusammenhang</b>	<b>373</b>
14.1 Wege: Definitionen und elementare Eigenschaften . . . . .	373
14.2 Erreichbarkeit und Zusammenhang . . . . .	378
14.3 Brücken und Schnittpunkte . . . . .	381

14.4	Kreisfreiheit und Bäume . . . . .	382
14.4.1	Kreisfreiheit und Zusammenhang . . . . .	382
14.4.2	a-Bäume . . . . .	384
14.4.3	f-Bäume . . . . .	386
14.5	Partielle Ordnung und Schichtennumerierung . . . . .	387
14.6	Klassifizierung von Zusammenhangskomponenten . . . . .	392
14.7	Abgeleitete Graphen . . . . .	394
14.8	Eulersche und hamiltonsche Wege . . . . .	399
14.8.1	Eulerwege . . . . .	399
14.8.1.1	a-Eulerkreise . . . . .	399
14.8.1.2	f-Eulerkreise . . . . .	400
14.8.2	Hamiltonwege . . . . .	404
14.9	Datenstrukturen für Wege und Kreiszerlegung . . . . .	405
14.9.1	Wege als verkettete Listen . . . . .	405
14.9.2	Kreiszerlegung* . . . . .	406
	Aufgaben . . . . .	408
	Literatur . . . . .	411
<b>15</b>	<b>Tiefensuche und Breitensuche</b>	<b>413</b>
15.1	Tiefensuche . . . . .	413
15.2	Tiefensuchbäume . . . . .	419
15.3	Bestimmung schwacher Zusammenhangskomponenten . . . . .	423
15.4	Bestimmung starker Zusammenhangskomponenten . . . . .	423
15.5	Breitensuche . . . . .	430
	Aufgaben . . . . .	433
	Literatur . . . . .	434
<b>16</b>	<b>Die Biblockzerlegung</b>	<b>437</b>
16.1	Klassen geschlossener a-Wege und Kantenzerlegungen . . . . .	437
16.2	Die Biblockzerlegung allgemeiner Graphen . . . . .	439
16.3	Eigenschaften der Komponenten der Biblockzerlegung . . . . .	443
16.4	Klassifikation von Linien und Knoten . . . . .	448
16.5	Der Biblockgraph . . . . .	449
16.6	Algorithmen zur Bestimmung der Biblockzerlegung . . . . .	451
16.7	Digraphen und vollständige Orientierungen . . . . .	458
	Aufgaben . . . . .	461
	Literatur . . . . .	462
<b>17</b>	<b>Perioden*</b>	<b>465</b>
17.1	a-Periode und f-Periode . . . . .	465
17.2	Periodizitätsklassen . . . . .	468

17.3	Ein Algorithmus zur Bestimmung der Periode . . . . .	470
	Aufgaben . . . . .	473
	Literatur . . . . .	475
<b>18</b>	<b>Ergänzungen zur Graphentheorie*</b>	<b>477</b>
18.1	Mengertheorie . . . . .	477
18.1.1	Trennende Mengen und disjunkte Wege . . . . .	477
18.1.2	Die Mengersätze . . . . .	478
18.1.3	Erweiterungen zu den Mengersätzen . . . . .	480
18.1.4	Die Struktur von Mengertrennmengen . . . . .	481
	Literatur zu Abschnitt „Mengertheorie“ . . . . .	482
18.2	Korrespondenzen . . . . .	482
18.2.1	Überdeckungen und Unabhängigkeit . . . . .	482
18.2.2	Korrespondenzen (Matchings) . . . . .	483
18.2.3	Maximale Korrespondenzen und alternierende Wege . . . . .	484
18.2.4	Ein Algorithmus zum Finden maximaler Korrespondenzen . . . . .	487
18.2.5	Korrektheit und Effizienz des Algorithmus . . . . .	492
18.2.6	Korrespondenzen in bipartiten Graphen . . . . .	494
	Literatur zu Abschnitt „Korrespondenzen“ . . . . .	495
18.3	Höhere Zusammenhangszerlegungen . . . . .	495
18.3.1	$k$ -a-Zusammenhang . . . . .	496
18.3.2	$k$ -a-Linienzusammenhang . . . . .	499
18.3.3	Beispiele . . . . .	500
	Aufgaben zu Abschnitt „Höhere Zusammenhangszerlegungen“ . . . . .	500
	Literatur zu Abschnitt „Höhere Zusammenhangszerlegungen“ . . . . .	500
18.4	Partitionen und starke Transitivität . . . . .	501
18.4.1	Partitionen in allgemeinen Graphen . . . . .	501
18.4.2	Starke Transitivität . . . . .	504
	Aufgaben zu Abschnitt „Partitionen und starke Transitivität“ . . . . .	505
	Literatur zu Abschnitt „Partitionen und starke Transitivität“ . . . . .	505
18.5	Knotenfärbungen . . . . .	505
	Aufgaben zu Abschnitt „Färbungen“ . . . . .	508
	Literatur zu Abschnitt „Färbungen“ . . . . .	508
18.6	Planarität . . . . .	509
	Literatur zu Abschnitt „Planarität“ . . . . .	510
<b>19</b>	<b>Kürzeste Wege in Netzwerken</b>	<b>511</b>
19.1	Gewichtete Wege . . . . .	511
19.2	Kürzeste Wege mit nichtnegativen Gewichten . . . . .	513
19.3	Kürzeste Wege mit beliebigen Gewichten . . . . .	516
19.4	Kürzeste Wege unter Berücksichtigung der Graphstruktur . . . . .	518

19.4.1	Kürzeste a-Entfernungen . . . . .	518
19.4.2	Kürzeste f-Entfernungen . . . . .	520
	Aufgaben . . . . .	523
	Literatur . . . . .	523
<b>20</b>	<b>Flüsse in Netzwerken*</b>	<b>525</b>
20.1	Definition und Zielsetzung . . . . .	525
20.2	Verbessernde Wege und Schnitte . . . . .	527
20.3	Algorithmus nach Ford und Fulkerson . . . . .	531
20.4	Der Algorithmus von Edmonds und Karp . . . . .	532
20.4.1	Beschreibung des Algorithmus . . . . .	532
20.4.2	Korrektheit und Komplexität des Algorithmus von Edmonds und Karp . . . . .	534
20.4.3	Vereinfachung des Netzes . . . . .	542
20.5	Nicht ganzzahlige Flüsse . . . . .	543
20.6	Existenz eines maximalen Flusses . . . . .	543
	Aufgaben . . . . .	544
	Literatur . . . . .	544
<b>21</b>	<b>Weitere Ergänzungen zu Netzwerken*</b>	<b>547</b>
21.1	Minimale erzeugende Bäume . . . . .	547
	Literatur zu Abschnitt „Minimale erzeugende Bäume“ . . . . .	549
21.2	Gewichtsoptimale Korrespondenzen . . . . .	549
	Literatur zu Abschnitt „Gewichtsoptimale Korrespondenzen“ . . . . .	551
21.3	Minimale Rundwege . . . . .	552
21.3.1	Das Problem des chinesischen Briefträgers . . . . .	552
21.3.2	Das Problem des Handlungsreisenden . . . . .	555
	Aufgaben zu Abschnitt „Minimale Rundwege“ . . . . .	555
	Literatur zu Abschnitt „Minimale Rundwege“ . . . . .	556
21.4	Endliche Markovketten . . . . .	556
21.4.1	Endliche Markovketten mit stationären Übergangswahrscheinlichkeiten . . . . .	556
21.4.2	Markovgraphen (Beispiel) . . . . .	557
21.4.3	Zustandsklassifizierung . . . . .	559
	Literatur zu Abschnitt „Endliche Markovketten“ . . . . .	562
<b>V</b>	<b>Parallelität</b>	<b>563</b>
<b>22</b>	<b>Parallelität in Rechensystemen und Netzen</b>	<b>565</b>
22.1	Parallelität auf der Hardware-Ebene . . . . .	565

22.2	Parallelität auf der Betriebssystem-Ebene . . . . .	567
22.3	Vernetzte Rechner . . . . .	576
22.4	Hochleistungsrechner . . . . .	577
	Literatur . . . . .	579
<b>23</b>	<b>Programmieren II: Parallele Programme</b>	<b>581</b>
23.1	Programmierbeispiele für Leichtgewichtsprozesse . . . . .	581
23.1.1	Beispiel: Paralleles Zählen . . . . .	582
23.1.2	Beispiel: Matrixmultiplikation . . . . .	585
23.1.3	Verallgemeinerter Speedup . . . . .	589
23.2	Erzeuger/Verbraucher . . . . .	590
23.2.1	Lösung mit Leichtgewichtsprozessen . . . . .	592
23.2.1.1	Programme . . . . .	592
23.2.1.2	Semaphore, Mutexe und Signale . . . . .	595
23.2.2	Lösung durch Simulation . . . . .	597
23.2.3	Bediensysteme* . . . . .	600
23.3	Beispiel: Zeigerspringen . . . . .	604
	Aufgaben . . . . .	607
	Literatur . . . . .	608
<b>VI</b>	<b>Anhänge</b>	<b>609</b>
<b>A</b>	<b>Mengenlehre</b>	<b>611</b>
A.1	Mengen . . . . .	611
A.2	Geordnete Paare und Relationen . . . . .	613
A.3	Abbildungen und Familien . . . . .	614
A.4	Allgemeine Vereinigungen, Durchschnitte und Produkte . . . . .	615
A.5	Relationen über einer Menge . . . . .	616
A.6	Mächtigkeit einer Menge . . . . .	616
<b>B</b>	<b>Wahrscheinlichkeitstheorie</b>	<b>619</b>
B.1	Allgemeines zu Wahrscheinlichkeitsräumen . . . . .	620
B.2	Diskrete Wahrscheinlichkeitsräume . . . . .	622
B.3	Stetige Wahrscheinlichkeitsräume . . . . .	624
B.4	Stochastische Prozesse . . . . .	626
B.5	Zufallszahlen . . . . .	626
	Literatur . . . . .	627

<b>C</b>	<b>Hilfsmittel aus der Analysis und Zahlentheorie</b>	<b>629</b>
C.1	Exponentialfunktion und Logarithmusfunktion . . . . .	629
C.2	Ungleichung von Cauchy-Schwarz-Bunjakowski . . . . .	631
C.3	Zahlentheoretische Hilfssätze . . . . .	632
<b>D</b>	<b>Ergänzungen zu Erzeuger/Verbraucher</b>	<b>635</b>
D.1	Erzeuger/Verbraucher in Realzeit . . . . .	635
D.2	Erzeuger/Verbraucher in der Simulation . . . . .	648
<b>E</b>	<b>Das System GHS zur Bearbeitung von Graphen</b>	<b>659</b>
E.1	Allgemeine Beschreibung von GHS . . . . .	659
E.2	Beispiel: Einlesen eines Graphen . . . . .	659
E.3	Beispiel: Schwache und starke Zusammenhangskomponenten . . . . .	660
E.4	Beispiel: Biblockzerlegung . . . . .	668
E.5	Beispiel: Finden von Hamiltonkreisen . . . . .	668
<b>F</b>	<b>Verzeichnis der Algorithmen</b>	<b>679</b>
<b>G</b>	<b>Namensliste</b>	<b>685</b>
<b>L</b>	<b>Lösungen ausgewählter Aufgaben</b>	<b>689</b>
L.1	Kapitel 5: „Algorithmen I“ . . . . .	689
L.2	Kapitel 6: „Algorithmen II“ . . . . .	690
L.3	Kapitel 9: „Suchbäume“ . . . . .	694
L.4	Kapitel 10: „Hashing“ . . . . .	695
L.5	Kapitel 11: „Sortieren“ . . . . .	695
L.6	Kapitel 12: „Grundlagen allgemeiner Graphen“ . . . . .	696
L.7	Kapitel 13: „Darstellung allgemeiner Graphen“ . . . . .	698
L.8	Kapitel 14: „Wege und Zusammenhang“ . . . . .	699
L.9	Kapitel 15: „Tiefen- und Breitensuche“ . . . . .	703
L.10	Kapitel 16: „Die Biblockzerlegung“ . . . . .	705
L.11	Kapitel 17: „Ergänzung: Perioden“ . . . . .	707
L.12	Kapitel 18: „Ergänzungen zur Graphentheorie“ . . . . .	709
L.13	Kapitel 19: „Kürzeste Wege in Netzwerken“ . . . . .	710
L.14	Kapitel 23 : „Programmieren II: Parallele Programme“ . . . . .	710
	<b>Abbildungsverzeichnis</b>	<b>711</b>
	<b>Tabellenverzeichnis</b>	<b>717</b>
	<b>Literaturverzeichnis</b>	<b>725</b>

*INHALTSVERZEICHNIS*

xv

**Stichwortverzeichnis**

**745**