

---

# Algorithmen kompakt und verständlich

---

Markus von Rimscha

# Algorithmen kompakt und verständlich

Lösungsstrategien am Computer

3., überarbeitete und erweiterte Auflage

Markus von Rimscha  
Fürth, Deutschland

Das in diesem Werk enthaltene Programm-Material ist mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Der Autor übernimmt infolgedessen keine Verantwortung und wird keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programm-Materials oder Teilen davon entsteht.

Verlag und Autor weisen darauf hin, dass keine Prüfung vorgenommen wurde, ob die Verwertung der im Buch beschriebenen Algorithmen und Verfahren mit Schutzrechten Dritter kollidiert. Verlag und Autor schließen insofern jede Haftung aus.

ISBN 978-3-658-05617-9  
DOI 10.1007/978-3-658-05618-6

ISBN 978-3-658-05618-6 (eBook)

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

Springer Vieweg

© Springer Fachmedien Wiesbaden 2008, 2010, 2014

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.

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier

Springer Vieweg ist eine Marke von Springer DE. Springer DE ist Teil der Fachverlagsgruppe Springer Science+Business Media.  
[www.springer-vieweg.de](http://www.springer-vieweg.de)

# Vorwort

Sowohl bei der praktischen Arbeit in der Software-Entwicklung, als auch im Rahmen meiner unterrichtenden Tätigkeit werde ich immer wieder gefragt, wie die eine oder andere Aufgabe am Rechner möglichst geschickt zu lösen sei. Selbstverständlich gibt es ein breites Sortiment an Fachbüchern zu quasi jedem Thema der Software-Entwicklung, also auch zu Problemlösungsstrategien am Computer – Algorithmen eben. Nicht selten umfassen diese jedoch hunderte von Seiten oder gar mehrere Bände, sind in Englisch geschrieben oder konzentrieren sich auf einen bestimmten Themenbereich wie etwa Verfahren der Künstlichen Intelligenz oder wiederum einen Teilaspekt wie Neuronale Netze. Der rote Faden geht hier all zu schnell verloren.

Im Ergebnis wünschen sich viele Software-Entwickler\* einen kompakten Leitfaden, um Probleme selbstständig anzugehen. Zwar sind zahlreiche Verfahren für konkrete Aufgaben bekannt und können im Zweifelsfall in der Literatur nachgeschlagen werden. Wenn aber eine neue oder sehr spezielle Herausforderung auftaucht, ist es an der Zeit, selbst eine passende Lösung zu entwickeln. Dazu muss die jeweilige Idee hinter den existierenden Verfahren bekannt sein. Deren Details bezüglich einzelner Aufgabenstellungen sind zunächst oft ebenso wenig wichtig wie das letzte Quäntchen an Optimierungspotenzial. Beides lässt sich meist nur bezogen auf die konkrete Aufgabe nutzen und ist damit von Mal zu Mal neu zu untersuchen.

Was möchte ich Ihnen also in diesem Buch nahe bringen? Wir werden uns hier unterschiedliche grundsätzliche Strategien ansehen, wie man am Computer Probleme lösen kann. Wir werden lernen, wie man das schnell und elegant tut, und ein Gefühl dafür entwickeln, ob ein Verfahren die passende Lösung für unsere Aufgabe ist oder nicht. Wir werden anhand von einfachen Beispielen unterschiedliche Strategien systematisch durchgehen, ihre Vor- und Nachteile kennen lernen und daraus passende Anwendungsgebiete ableiten.

Bewusst werden wir uns mit Beispielen aus unterschiedlichsten Bereichen beschäftigen, seien es Spielstrategien, Gewinnmaximierung durch Optimierung oder Mustererkennung mit Hilfe Künstlicher Intelligenz. Wir werden uns keines der hier behandelten Themen in voller Tiefe ansehen – dazu sei auf die einschlägige Literatur verwiesen. Statt dessen gehen wir nur so weit ins Detail, wie es nötig ist, um ein konkretes und lauffähiges Beispiel nachzuvollziehen. Also wird es auch nicht unser Wunsch sein, am Ende eine tausendseitige Abtipp-Vorlage für alle Lebenslagen in den Händen zu halten. Vielmehr möchten wir uns mit dem nötigen

---

\* Aus Gründen der Lesbarkeit wird in diesem Buch nicht zwischen der männlichen und weiblichen Form unterschieden; es sind jeweils beide Geschlechter gemeint.

Handwerkszeug versorgen, um Probleme künftig selbstständig anzugehen. Am Ende werden wir in der Lage sein, Aufgaben am Computer zu lösen, an denen wir bisher vielleicht gescheitert sind.

Ich wende mich mit diesem Buch an alle, die bereits erste Erfahrungen in der Programmierung gesammelt haben und wissen, wie einfache Funktionen zu programmieren sind. Wir werden uns hier einige wichtige Methoden in Form von Programmcode ansehen, ausführlichere Beispiele finden sich im Online-Bereich dieses Buchs unter <http://www.springer.com/978-3-658-05617-9>. Um den roten Faden nicht aus den Augen zu verlieren, werden wir auf viele Fehlerabfragen etc. verzichten, die zwar sinnvoll wären, den Code aber unübersichtlich machen.

Mein Dank gilt meinen Eltern und allen, die mich beim Schreiben dieses Buches unterstützt haben, insbesondere Robert.

Ich wünsche Ihnen nun viel Spaß beim Lesen und Erfolg bei der Umsetzung!

Markus von Rimscha

Juli 2008

### **Vorwort zur zweiten Auflage**

Ich habe mich sehr über das rege Interesse an diesem Buch gefreut.

Dank des konstruktiven Feedbacks und wertvoller Anregungen konnte ich Tippfehler beseitigen und habe einige Passagen und Code-Beispiele ergänzt.

Ich wünsche Ihnen weiterhin viel Spaß beim Lesen und Erfolg bei der Umsetzung!

Markus von Rimscha

September 2009

### **Vorwort zur dritten Auflage**

Über das nach wie vor große Interesse an meinem Buch freue ich mich sehr.

Neben verschiedenen Ergänzungen, insbesondere im Bereich des Multi-Threading, habe ich auf vielfachen Wunsch nun auch einige Code-Beispiele in C# aufgenommen, die im Online-Bereich unter <http://www.springer.com/978-3-658-05617-9> zu finden sind. Der Schwerpunkt liegt jedoch nach wie vor auf Java; auch im Text bin ich Java treu geblieben.

Weiterhin wünsche ich Ihnen nun viel Spaß beim Lesen und natürlich Erfolg bei der praktischen Anwendung.

---

Gleichzeitig möchte ich Sie ermuntern, sich nicht nur mit technischen Herausforderungen sondern auch mit der gesellschaftlichen Dimension der Software-Entwicklung zu beschäftigen. Wenn wir in diesem Buch „Algorithmen“ untersuchen, dann geht es in der Regel um die Arbeit mit Daten. Deren Erhebung und Verarbeitung sollte – so wie vor allem auch das Ziehen von Schlussfolgerungen daraus – in einem vertretbaren und kritisch hinterfragen Umfang erfolgen, unter Wahrung sowohl gesellschaftlicher als insbesondere auch individueller Interessen.

Wissen ist Macht – das gilt heute mehr denn je. Die aus der Gewinnung und Verarbeitung von Daten resultierende Macht muss von demokratisch gewählten Institutionen und insbesondere auch von einer kritischen Öffentlichkeit kontrolliert werden. Auch dazu ist Wissen nötig. Sich selbst bzw. andere künstlich dumm zu halten, kann nicht die Lösung sein – und ist auch kaum möglich. Wissen kann nicht zurückgenommen werden und ein Blick über den Tellerrand schadet nicht.

Nun können wir nicht nur anderen auf die Finger sehen, wir können auch selbst Software schreiben – darum geht es in diesem Buch. Wir müssen dabei jedoch nicht alles programmieren, nur weil wir es *können* – oder zu können glauben.

In beiden Fällen sollten wir im Hinterkopf behalten, dass niemand frei von Fehlern und Schwächen ist: nicht die Computer und schon gar nicht die Menschen, die sie programmieren. Um das zu erkennen, brauchen wir uns nur anzusehen, mit welcher Häufigkeit kritische Sicherheitslücken in weit verbreiteter Software aufgespürt und leider auch ausgenutzt werden. Menschen machen nicht nur Fehler; sie haben auch ihre eigenen Interessen, die nicht immer freundlicher Natur sind. Ein Computer berechnet das, was ein Mensch ihm sagt. Ein Computer verarbeitet Daten, die ein Mensch ihm gibt. Hier „Objektivität“ zu vermuten, ist blanke Illusion. Dementsprechend blauäugig ist es, sich blind auf die Technik zu verlassen. Je mehr unsere Computer *arbeiten*, desto mehr müssen wir selbst *denken*. Deswegen sollten wir jede Aussage, die ein Rechner über uns trifft, kritisch hinterfragen – insbesondere, wenn es sich um statistische Aussagen handelt. Diese sind im besten Fall richtig berechnet und damit *statistisch* korrekt, erlauben aber keine *individuelle* Wertung. „Wenn Deine Freunde ..., dann bist auch Du ...“ ist ein Vorurteil, wie es klassischer kaum sein könnte. Eine solche Denke ist gefährlich und wird ganz gewiss nicht dadurch besser, dass sie auch noch programmiert wird. Es ist nun einmal ein Unterschied, ob wir die Anzahl der Kinobesucher vorhersagen oder einzelne Menschen mit wertenden Aussagen konfrontieren – nur weil wir meinen, zu wissen, was in deren Köpfen vorgeht und wie sie sich *in Zukunft* verhalten werden. Wir wissen es nicht, und das ist meistens auch gut so.

Wir sollten uns weder das selbstständige und kritische Denken abgewöhnen bzw. abgewöhnen lassen, noch sollten wir all zu technik-verliebt an Unfehlbarkeit glauben, die es nirgendwo gibt – zum Glück, möchte man meinen.

Nun sollen diese Überlegungen sicherlich kein Plädoyer gegen den technischen Fortschritt sein. In diesem Fall dürften die nächsten knapp 200 Seiten überhaupt nicht existieren. Den Fortschritt aufhalten zu wollen, wäre albern. Dessen Nutzen

sowohl auf gesellschaftlicher als auch auf individueller Ebene zu bestreiten, würde von einer höchst selektiven Wahrnehmung zeugen. Auch sollten wir nicht, unter dauernder Paranoia leidend, nur noch Gefahren und Risiken sehen, dabei Chancen ignorieren und ständig mit erhobenem Zeigefinger alles und jeden kritisieren – außer uns selbst natürlich. Das steht uns schlicht und einfach nicht zu und nicht zuletzt stellen wir damit Ansprüche an den Rest der Welt, die wir selbst niemals erfüllen könnten. Kompromisse sind unvermeidlich, die Welt ist nicht schwarz oder weiss sondern meistens grau, so wie wir selbst. Wir brauchen neue Technologien – insbesondere Computer. Ohne sie funktioniert eine moderne Gesellschaft nicht mehr. Ohne sie wäre auch unser Wohlstand kaum darstellbar. Wir sollten neue Entwicklungen also gewiss nicht verteufeln oder aus Prinzip ablehnen, wir sollten sie aber aufmerksam und kritisch verfolgen und hinterfragen – das hat nichts mit hinterwäldlerischer Technik-Feindlichkeit zu tun.

„Früher war alles besser, sogar die Zukunft. Die Menschen waren glücklich – auch ohne diesen ganzen technischen Unrat. Wir brauchen diesen Kram nicht, weg damit!“ Das ist ebenso falsch wie arrogant und feige. Falsch, denn mit dieser Einstellung hätten wir wohl noch immer nicht das Rad erfunden. Arrogant, denn wir machen uns selbst zum Maß der Dinge und ignorieren die Wünsche und Befindlichkeiten anderer. Feige, denn statt andere Sichtweisen wenigstens wahrzunehmen und uns der Diskussion zu stellen, tun wir ganz still und leise so, als gäbe es gar keine anderen Ansichten – zumindest keine beachtenswerten – und drücken uns damit von vorne herein vor jeder ernsthaften Auseinandersetzung.

„Alles Neue ist toll! Mitmachen! Aktiv sein! Am besten gleich proaktiv! Wer Zweiter wird, verliert. Lieber mitgemacht als nachgedacht. Wir wissen schliesslich, wo's langgeht. Wer kritisch nachfragt, hat offenbar etwas zu verbergen. Auf die Bedenken anderer kann keine Rücksicht genommen werden. Los geht's!“ Gilt dafür nicht genau das Gleiche? Falsch. Arrogant. Feige.

Wir brauchen Computer, sollten ihnen aber nicht das Denken überlassen. Denkfaul zu sein, ist schlimm genug. Dies auch noch zur Tugend zu erklären, führt in eine Sackgasse. In unserem Verhältnis zu Computern sollten wir nicht vergessen, wer wen erfunden hat und wer wem dienen sollte.

Markus von Rimscha

Mai 2014

# Inhalt

1 Einführung .....	1
2 Arten von Algorithmen .....	3
2.1 Iterative Algorithmen .....	5
2.1.1 Sortieren .....	6
2.1.2 Wege im Labyrinth .....	8
2.1.3 Bewertung .....	12
2.2 Rekursive Algorithmen .....	13
2.2.1 Die Türme von Hanoi .....	15
2.2.2 Sortieren .....	19
2.2.3 Spielalgorithmen und Schach .....	21
2.2.4 Fraktale und Bildkompression .....	27
2.2.5 Auswertung von Formelausdrücken .....	35
2.2.6 Bewertung .....	40
2.3 Dynamische Algorithmen .....	43
2.3.1 Fibonacci-Zahlen .....	43
2.3.2 Bewertung .....	46
2.4 Heuristische Algorithmen .....	47
2.4.1 Sortieren .....	48
2.4.2 Bewertung .....	56
2.5 Zufallsgesteuerte Algorithmen .....	57
2.5.1 Metropolis-Algorithmus und Simulated Annealing .....	57
2.5.2 Zufallsgesteuerte Erweiterung bestehender Algorithmen .....	62
2.5.3 Bewertung .....	63
2.6 Genetische Algorithmen .....	65
2.6.1 Rucksack-Problem .....	67
2.6.2 Gewinnmaximierung .....	70
2.6.3 Bewertung .....	74
2.7 Probabilistische Algorithmen .....	75
2.7.1 Multiplikationstest .....	76
2.7.2 Primzahltest .....	78
2.7.3 Bewertung .....	80



3	Effizienz eines Algorithmus .....	81
3.1	Wachstum .....	82
3.2	Bewertung eines Algorithmus .....	87
3.2.1	Average-Case und Worst-Case .....	89
3.2.2	Minimaler Aufwand .....	90
3.3	Laufzeit und Speicher .....	92
3.4	Parallele Verarbeitung .....	93
3.4.1	Parallele Algorithmen .....	93
3.4.2	Parallele Programmierung .....	95
3.5	Übersicht .....	112
3.6	Nutzung praktisch unlösbarer Probleme und Verschlüsselung .....	114
4	Wichtige Datenstrukturen .....	125
4.1	Listen .....	126
4.2	Mengen .....	128
4.2.1	Sortierte Mengen .....	128
4.2.2	Unsortierte Mengen .....	128
4.3	Zuordnungen .....	130
4.4	Bäume .....	131
4.5	Graphen .....	133
5	Künstliche Intelligenz .....	139
5.1	Maschinelles Lernen .....	143
5.1.1	Entscheidungsbäume .....	143
5.1.2	Bewertung .....	158
5.2	Schwarmintelligenz .....	159
5.2.1	Ameisenalgorithmen .....	159
5.2.2	Bewertung .....	170
5.3	Neuronale Netze .....	171
5.3.1	Hebb'sche Regel .....	174
5.3.2	Backpropagation .....	176
5.3.3	Erweiterungen .....	179
5.3.4	Bewertung .....	184
	Literaturverzeichnis .....	185
	Stichwortverzeichnis .....	191