

# Algorithmen

**Algorithmen und Datenstrukturen**

4., aktualisierte Auflage

**Robert Sedgewick  
Kevin Wayne**

# Inhaltsverzeichnis

<b>Vorwort</b>	11
Besondere Merkmale .....	11
Die Website zum Buch .....	13
Das Buch als Unterrichtsmittel .....	14
Kontext .....	14
Danksagung .....	15
<b>Vorwort zur deutschen Ausgabe</b>	16
Webinhalte zum vorliegenden Buch .....	17
<b>Kapitel 1 Grundlagen</b>	19
Algorithmen .....	20
Zusammenfassung der Themen .....	23
1.1 Das grundlegende Programmiermodell .....	25
1.1.1 Grundlegende Struktur eines Java-Programms .....	27
1.1.2 Primitive Datentypen und Ausdrücke .....	28
1.1.3 Anweisungen .....	31
1.1.4 Kurzschreibweisen .....	34
1.1.5 Arrays .....	36
1.1.6 Statische Methoden .....	39
1.1.7 APIs .....	47
1.1.8 Strings .....	52
1.1.9 Ein- und Ausgabe .....	54
1.1.10 Binäre Suche .....	65
1.1.11 Ausblick .....	69
1.2 Datenabstraktion .....	81
1.2.1 Abstrakte Datentypen .....	82
1.2.2 Beispiele abstrakter Datentypen .....	92
1.2.3 Abstrakte Datentypen implementieren .....	104
1.2.4 Weitere Implementierungen abstrakter Datentypen .....	110
1.2.5 Datentypdesign .....	116
1.3 Multimengen, Warteschlangen und Stapel .....	139
1.3.1 APIs .....	140
1.3.2 Collections implementieren .....	151
1.3.3 Verkettete Listen .....	162
1.3.4 Zusammenfassung .....	176

1.4	Analyse der Algorithmen	191
1.4.1	Die wissenschaftliche Methode	191
1.4.2	Beobachtungen	192
1.4.3	Mathematische Modelle	198
1.4.4	Klassifikationen der Wachstumsordnung	206
1.4.5	Schnellere Algorithmen entwerfen	209
1.4.6	Experimente zum Verdopplungsverhältnis	212
1.4.7	Fallstricke	216
1.4.8	Die Abhängigkeit von Eingaben reduzieren	218
1.4.9	Speicherbedarf	221
1.4.10	Ausblick	227
1.5	Fallstudie Union-Find	238
1.5.1	Verwaltung von Zusammenhangskomponenten	238
1.5.2	Implementierungen	245
1.5.3	Ausblick	257
<b>Kapitel 2</b>	<b>Sortieren</b>	<b>265</b>
2.1	Elementare Sortierverfahren	267
2.1.1	Spielregeln	267
2.1.2	Selektionsort	272
2.1.3	Insertionsort	274
2.1.4	Sortieralgorithmen grafisch darstellen	276
2.1.5	Zwei Sortieralgorithmen vergleichen	277
2.1.6	Shellsort	281
2.2	Mergesort	294
2.2.1	Abstraktes In-Place-Mergen	294
2.2.2	Top-Down-Mergesort	296
2.2.3	Bottom-Up-Mergesort	301
2.2.4	Die Komplexität des Sortierens	304
2.3	Quicksort	313
2.3.1	Der grundlegende Algorithmus	313
2.3.2	Laufzeitverhalten	318
2.3.3	Algorithmische Verbesserungen	320
2.4	Vorrangwarteschlangen	333
2.4.1	API	334
2.4.2	Einfache Implementierungen	336
2.4.3	Heap-Definitionen	339
2.4.4	Algorithmen für Heaps	341
2.4.5	Heapsort	350
2.5	Anwendungen	363
2.5.1	Verschiedene Datentypen sortieren	364
2.5.2	Welchen Sortieralgorithmus soll ich verwenden?	369
2.5.3	Reduktionen	372
2.5.4	Sortieranwendungen im kurzen Überblick	375

<b>Kapitel 3</b>	<b>Suchen</b>	<b>387</b>
3.1	Symboltabellen . . . . .	389
3.1.1	API . . . . .	390
3.1.2	Geordnete Symboltabellen . . . . .	393
3.1.3	Beispielclients . . . . .	398
3.1.4	Sequenzielle Suche in einer ungeordneten verketteten Liste . . . .	402
3.1.5	Binäre Suche in einem geordneten Array . . . . .	405
3.1.6	Analyse der binären Suche . . . . .	411
3.1.7	Ausblick . . . . .	413
3.2	Binäre Suchbäume . . . . .	424
3.2.1	Grundlegende Implementierung . . . . .	425
3.2.2	Analyse . . . . .	432
3.2.3	Ordnungsbasierte Methoden und Löschen . . . . .	435
3.3	Balancierte Suchbäume . . . . .	453
3.3.1	2-3-Suchbäume . . . . .	453
3.3.2	Rot-Schwarz-Bäume . . . . .	461
3.3.3	Implementierung . . . . .	470
3.3.4	Löschen . . . . .	473
3.3.5	Eigenschaften von Rot-Schwarz-Bäumen . . . . .	475
3.4	Hashtabellen . . . . .	489
3.4.1	Hashfunktionen . . . . .	490
3.4.2	Hashing mit Verkettung . . . . .	496
3.4.3	Hashing mit linearer Sondierung . . . . .	501
3.4.4	Größenanpassung von Arrays . . . . .	506
3.4.5	Speicher . . . . .	509
3.5	Anwendungen . . . . .	519
3.5.1	Welche Symboltabellen-Implementierung soll ich verwenden? . .	519
3.5.2	Mengen-APIs (Set) . . . . .	522
3.5.3	Wörterbuch-Anwendungen . . . . .	526
3.5.4	Indizierungsclients . . . . .	531
3.5.5	Dünn besetzte Vektoren . . . . .	537
<b>Kapitel 4</b>	<b>Graphen</b>	<b>549</b>
4.1	Ungerichtete Graphen . . . . .	553
4.1.1	Glossar . . . . .	554
4.1.2	Datentyp für ungerichtete Graphen . . . . .	557
4.1.3	Tiefensuche . . . . .	566
4.1.4	Pfadsuche . . . . .	572
4.1.5	Breitensuche . . . . .	577
4.1.6	Zusammenhangskomponenten . . . . .	583
4.1.7	Symbolgraphen . . . . .	589
4.1.8	Zusammenfassung . . . . .	597

4.2	Gerichtete Graphen . . . . .	607
4.2.1	Glossar . . . . .	607
4.2.2	Datentyp für Digraphen . . . . .	609
4.2.3	Erreichbarkeit in Digraphen . . . . .	612
4.2.4	Zyklen und azyklische Digraphen. . . . .	617
4.2.5	Starker Zusammenhang in Digraphen. . . . .	628
4.2.6	Zusammenfassung . . . . .	638
4.3	Minimale Spannbäume . . . . .	646
4.3.1	Zugrunde liegende Prinzipien. . . . .	648
4.3.2	Datentyp eines kantengewichteten Graphen . . . . .	651
4.3.3	API und Testclient für minimale Spannbäume . . . . .	655
4.3.4	Der Algorithmus von Prim. . . . .	658
4.3.5	Eager-Version des Prim-Algorithmus . . . . .	663
4.3.6	Der Algorithmus von Kruskal . . . . .	668
4.3.7	Ausblick . . . . .	671
4.4	Kürzeste Pfade . . . . .	680
4.4.1	Eigenschaften der kürzeste Pfade . . . . .	682
4.4.2	Datentypen für kantengewichtete Digraphen . . . . .	684
4.4.3	Theoretische Grundlagen für Kürzeste-Pfade-Algorithmen . . . . .	692
4.4.4	Algorithmus von Dijkstra. . . . .	694
4.4.5	Azyklische kantengewichtete Digraphen . . . . .	701
4.4.6	Kürzeste Pfade in allgemeinen kantengewichteten Digraphen . . . . .	711
4.4.7	Ausblick . . . . .	726
<b>Kapitel 5</b>	<b>Strings</b> . . . . .	<b>737</b>
5.1	Stringsortierverfahren . . . . .	745
5.1.1	Schlüsselindiziertes Zählen . . . . .	746
5.1.2	LSD-Sortierverfahren . . . . .	749
5.1.3	MSD-Sortierverfahren . . . . .	752
5.1.4	3-Wege-Quicksort für Strings . . . . .	762
5.1.5	Welchen Stringsortalgorithmus soll ich verwenden? . . . . .	767
5.2	Tries . . . . .	773
5.2.1	Tries . . . . .	775
5.2.2	Eigenschaften von Tries. . . . .	785
5.2.3	Ternäre Suchtries . . . . .	789
5.2.4	TST-Eigenschaften . . . . .	792
5.2.5	Welche Symboltabellen-Implementierung soll ich für Strings verwenden? . . . . .	795

5.3	Teilstringsuche	800
5.3.1	Ein kurzer geschichtlicher Abriss	800
5.3.2	Brute-Force-Teilstringsuche	801
5.3.3	Teilstringsuche nach Knuth-Morris-Pratt	804
5.3.4	Teilstringsuche nach Boyer-Moore	812
5.3.5	Fingerprint-Suche nach Rabin-Karp	817
5.3.6	Zusammenfassung	822
5.4	Reguläre Ausdrücke	829
5.4.1	Muster mit regulären Ausdrücken	830
5.4.2	Abkürzungen	832
5.4.3	Reguläre Ausdrücke in Anwendungen	834
5.4.4	Nichtdeterministische endliche Automaten	836
5.4.5	Simulation eines NEA	839
5.4.6	Konstruktion eines NEA für einen regulären Ausdruck	842
5.5	Datenkomprimierung	851
5.5.1	Spielregeln	852
5.5.2	Binärdaten lesen und schreiben	853
5.5.3	Beschränkungen	857
5.5.4	Aufwärmübung: Genomik	860
5.5.5	Lauf längencodierung	863
5.5.6	Huffman-Komprimierung	868
5.5.7	LZW-Komprimierung	882
<b>Kapitel 6</b>	<b>Im Kontext</b>	<b>895</b>
	Ereignisgesteuerte Simulation	899
	B-Bäume	909
	Suffixarrays	918
	Netzwerkflussalgorithmen	928
	Reduktion	946
	Nicht effizient lösbare Probleme	953
	Allgemeine Übungen zu der Kollisionssimulation	966
	Allgemeine Übungen zu B-Bäumen	968
	Allgemeine Übungen zu Suffixarrays	969
	Allgemeine Übungen zu Max-Fluss	971
	Allgemeine Übungen zu Reduktionen und scheinbarer Unlösbarkeit	973
<b>Register</b>		<b>975</b>