

Inhaltsverzeichnis

1 Grundlagen	1
1.1 Algorithmen und ihre formalen Eigenschaften	1
1.2 Beispiele arithmetischer Algorithmen	5
1.2.1 Ein Multiplikationsverfahren	5
1.2.2 Polynomprodukt	8
1.2.3 Schnelle Multiplikation von Zahlen und von Matrizen	11
1.2.4 Polynomprodukt und <i>FFT</i>	15
1.3 Verschiedene Algorithmen für dasselbe Problem	20
1.4 Die richtige Wahl einer Datenstruktur	24
1.5 Lineare Listen	29
1.5.1 Sequenzielle Speicherung linearer Listen	31
1.5.2 Verkettete Speicherung linearer Listen	33
1.5.3 Stapel und Schlangen	41
1.6 Ausblick auf weitere Datenstrukturen	48
1.7 Skip-Listen	50
1.7.1 Perfekte und randomisierte Skip-Listen	51
1.7.2 Analyse	57
1.8 Implementation von Datenstrukturen und Algorithmen in Java	60
1.8.1 Einige Elemente von Java	61
1.8.2 Implementation linearer Listen	62
1.9 Aufgaben	68
2 Sortieren	79
2.1 Elementare Sortierverfahren	82
2.1.1 Sortieren durch Auswahl	82
2.1.2 Sortieren durch Einfügen	85
2.1.3 Shellsort	88
2.1.4 Bubblesort	89
2.2 Quicksort	92
2.2.1 Quicksort: Sortieren durch rekursives Teilen	93
2.2.2 Quicksort-Varianten	102
2.3 Heapsort	106
2.4 Mergesort	112

2.4.1	2-Wege-Mergesort	113
2.4.2	Reines 2-Wege-Mergesort	116
2.4.3	Natürliches 2-Wege-Mergesort	118
2.5	Radixsort	121
2.5.1	Radix-exchange-sort	121
2.5.2	Sortieren durch Fachverteilung	123
2.6	Sortieren vorsortierter Daten	127
2.6.1	Maße für Vorsortierung	128
2.6.2	A-sort	132
2.6.3	Sortieren durch lokales Einfügen und natürliches Verschmelzen	137
2.7	Externes Sortieren	141
2.7.1	Das Magnetband als Externspeichermedium	142
2.7.2	Ausgeglichenes 2-Wege-Mergesort	144
2.7.3	Ausgeglichenes Mehr-Wege-Mergesort	147
2.7.4	Mehrphasen-Mergesort	151
2.8	Untere Schranken	153
2.9	Implementation und Test von Sortierverfahren in Java	158
2.10	Aufgaben	162
3	Suchen	167
3.1	Das Auswahlproblem	168
3.2	Suchen in sequenziell gespeicherten linearen Listen	173
3.2.1	Sequenzielle Suche	173
3.2.2	Binäre Suche	174
3.2.3	Fibonacci-Suche	176
3.2.4	Exponentielle Suche	179
3.2.5	Interpolationssuche	180
3.3	Selbstanordnende lineare Listen	180
3.4	Java Implementation	186
3.5	Aufgaben	188
4	Hashverfahren	191
4.1	Zur Wahl der Hashfunktion	193
4.1.1	Die Divisions-Rest-Methode	193
4.1.2	Die multiplikative Methode	194
4.1.3	Perfektes und universelles Hashing	194
4.2	Hashverfahren mit Verkettung der Überläufer	198
4.3	Offene Hashverfahren	203
4.3.1	Lineares Sondieren	205
4.3.2	Quadratisches Sondieren	207
4.3.3	Uniformes und zufälliges Sondieren	208
4.3.4	Double Hashing	211
4.3.5	Ordered Hashing	215
4.3.6	Robin-Hood-Hashing	220
4.3.7	Coalesced Hashing	221
4.4	Dynamische Hashverfahren	225
4.4.1	Lineares Hashing	227

4.4.2	Virtuelles Hashing	232
4.4.3	Erweiterbares Hashing	236
4.5	Das Gridfile	239
4.6	Implementation von Hashverfahren in Java	249
4.7	Aufgaben	254
5	Bäume	259
5.1	Natürliche Bäume	262
5.1.1	Suchen, Einfügen und Entfernen von Schlüsseln	266
5.1.2	Durchlaufordnungen in Binärbäumen	272
5.1.3	Analytische Betrachtungen	275
5.2	Balancierte Binärbäume	284
5.2.1	AVL-Bäume	284
5.2.2	Bruder-Bäume	296
5.2.3	Gewichtsbalancierte Bäume	311
5.3	Randomisierte Suchbäume	318
5.3.1	Treaps	319
5.3.2	Treaps mit zufälligen Prioritäten	321
5.4	Selbstanordnende Binärbäume	327
5.4.1	Splay-Bäume	328
5.4.2	Amortisierte Worst-case-Analyse	332
5.5	B-Bäume	339
5.5.1	Suchen, Einfügen und Entfernen in B-Bäumen	344
5.6	Weitere Klassen	349
5.6.1	Übersicht	349
5.6.2	Konstante Umstrukturierungskosten und relaxiertes Balancieren	354
5.6.3	Eindeutig repräsentierte Wörterbücher	371
5.7	Optimale Suchbäume	377
5.8	Alphabetische und mehrdimensionale Suchbäume	383
5.8.1	Tries	384
5.8.2	Quadranten- und 2d-Bäume	385
5.9	Implementation von Bäumen und dazugehöriger Algorithmen in Java	389
5.10	Aufgaben	394
6	Manipulation von Mengen	403
6.1	Vorrangwarteschlangen	404
6.1.1	Dijkstras Algorithmus zur Berechnung kürzester Wege	405
6.1.2	Implementation von Priority Queues mit verketteten Listen und balancierten Bäumen	408
6.1.3	Linksbäume	410
6.1.4	Binomial Queues	413
6.1.5	Fibonacci-Heaps	420
6.2	Union-Find-Strukturen	428
6.2.1	Kruskals Verfahren zur Berechnung minimaler spannender Bäume	428
6.2.2	Vereinigung nach Größe und Höhe	431
6.2.3	Methoden der Pfadverkürzung	435

6.3	Allgemeiner Rahmen	439
6.4	Aufgaben	442
7	Weitere Algorithmenentwurfstechniken	445
7.1	Ein einfaches Beispiel: Fibonacci-Zahlen	446
7.2	Erreichbare Teilsumme	447
7.2.1	Eine einfache Lösung	447
7.2.2	Eine bessere Lösung	448
7.2.3	Eine Lösung <i>von unten nach oben</i>	449
7.3	Das Rucksackproblem	452
7.3.1	Eine exakte Lösung <i>von unten nach oben</i>	452
7.3.2	Eine Familie von Näherungslösungen	453
7.3.3	Das Optimalitätsprinzip der Dynamischen Programmierung	456
7.4	Längste gemeinsame Teilfolge	456
7.5	Das Backtrack-Prinzip	457
7.5.1	Ein Beispiel: Das Vier-Damen-Problem	457
7.5.2	Die Lösung als rekursive Prozedur	458
7.5.3	Formale Fassung des Prinzips als Programmrahmen	460
7.5.4	Anwendung auf weitere Probleme	461
7.5.5	Erweiterungen	464
7.6	Aufgaben	465
8	Geometrische Algorithmen	471
8.1	Einleitung	471
8.2	Die konvexe Hülle	472
8.2.1	Jarvis' Marsch	474
8.2.2	Graham's Scan	475
8.2.3	Linearer Scan	478
8.3	Das Scan-line-Prinzip	478
8.3.1	Sichtbarkeitsproblem	480
8.3.2	Das Schnittproblem für iso-orientierte Liniensegmente	483
8.3.3	Das allgemeine Liniensegment-Schnittproblem	486
8.4	Geometrisches Divide-and-conquer	492
8.4.1	Segmentschnitt mittels Divide-and-conquer	493
8.4.2	Inklusions- und Schnittprobleme für Rechtecke	498
8.5	Geometrische Datenstrukturen	501
8.5.1	Reduktion des Rechteckschnittproblems	502
8.5.2	Segment-Bäume	505
8.5.3	Intervall-Bäume	512
8.5.4	Prioritäts-Suchbäume	515
8.6	Anwendungen geometrischer Datenstrukturen	529
8.6.1	Ein Spezialfall des HLE-Problems	530
8.6.2	Dynamische Bereichssuche mit einem festen Fenster	537
8.7	Distanzprobleme und ihre Lösung	540
8.7.1	Distanzprobleme	541
8.7.2	Das Voronoi-Diagramm	545
8.7.3	Die Speicherung des Voronoi-Diagramms	550

8.7.4	Die Konstruktion des Voronoi-Diagramms	553
8.7.5	Lösungen für Distanzprobleme	559
8.8	Das Nächste-Punkte-Paar-Problem	569
8.8.1	Scan-line-Lösung für das CP-Problem	569
8.8.2	Divide-and-conquer-Lösung für das CP-Problem	572
8.8.3	Ein randomisiertes Verfahren zur Lösung des CP-Problem	575
8.9	Aufgaben	580
9	Graphenalgorithmien	589
9.1	Topologische Sortierung	597
9.2	Transitive Hülle	600
9.2.1	Transitive Hülle allgemein	600
9.2.2	Transitive Hülle für azyklische Digraphen	602
9.3	Durchlaufen von Graphen	604
9.3.1	Einfache Zusammenhangskomponenten	607
9.3.2	Strukturinformation durch Tiefensuche	607
9.4	Zusammenhangskomponenten	611
9.4.1	Zweifache Zusammenhangskomponenten	611
9.4.2	Starke Zusammenhangskomponenten	615
9.5	Kürzeste Wege	619
9.5.1	Kürzeste Wege in Distanzgraphen	620
9.5.2	Kürzeste Wege in beliebig bewerteten Graphen	625
9.5.3	Alle kürzesten Wege	629
9.6	Minimale spannende Bäume	631
9.7	Flüsse in Netzwerken	637
9.8	Zuordnungsprobleme	648
9.8.1	Maximale Zuordnungen in bipartiten Graphen	650
9.8.2	Maximale Zuordnungen im allgemeinen Fall	653
9.8.3	Maximale gewichtete Zuordnungen	661
9.9	Aufgaben	662
10	Suchen in Texten	669
10.1	Suchen in dynamischen Texten	670
10.1.1	Das naive Verfahren zur Textsuche	670
10.1.2	Das Verfahren von Knuth-Morris-Pratt	672
10.1.3	Das Verfahren von Boyer-Moore	676
10.1.4	Signaturen	682
10.2	Approximative Zeichenkettensuche	683
10.3	Suchen in statischen Texten (gemeinsam mit S. Schuierer)	695
10.3.1	Aufbereitung von Texten – Suffix-Bäume	695
10.3.2	Analyse	703
10.4	Aufgaben	706

11 Ausgewählte Themen	709
11.1 Randomisierte Algorithmen	709
11.1.1 Randomisiertes Quicksort	710
11.1.2 Randomisierter Primzahltest	712
11.1.3 Öffentliche Verschlüsselungssysteme	716
11.2 Parallele Algorithmen	722
11.2.1 Einfache Beispiele paralleler Algorithmen	723
11.2.2 Paralleles Mischen und Sortieren	729
11.2.3 Systolische Algorithmen	740
11.3 Aufgaben	743
Literaturverzeichnis	747