

Inhaltsverzeichnis

1	Grundlegende Begriffe	11
1.1	Der Begriff des „Algorithmus“	11
1.2	Pseudocode	12
1.3	Fehlerbehandlung	13
1.4	Abstrakte Datentypen	13
1.4.1	Schnittstelle eines ADT	14
1.4.2	Vor- und Nachbedingungen	14
1.4.3	Invariante eines ADT	15
1.4.4	Implementation	16
1.4.5	Notation von ADTs	16
1.4.6	Beispiele	17
2	Komplexität von Algorithmen	21
2.1	Asymptotische Analyse	22
2.1.1	Die O -Notation	22
2.1.2	Die Ω -Notation	24
2.1.3	Beispiel	25
2.1.4	Die Θ -Notation	26
2.2	Häufig auftretende Ressourcenanforderungen	26
2.3	Analyse rekursiver Algorithmen	28
2.3.1	Raten und Beweisen	29
2.4	Zusammenfassung	31

3	Elementare Datenstrukturen	33
3.1	Lineare und kreisförmige Strukturen	33
3.1.1	Einfach und doppelt verkettete Listen	33
3.1.2	Ringlisten	36
3.1.3	Darstellung höherer Strukturen	37
3.2	Graphen und Bäume	38
3.2.1	Grundlegende Begriffe	38
3.2.2	Bäume und ihre Darstellung	39
3.2.3	Darstellung von Graphen	44
4	Internes Sortieren	47
4.1	Elementare Sortiervverfahren	48
4.1.1	Sortieren durch Auswählen I: Selection Sort	48
4.1.2	Sortieren durch Einfügen: Insertion Sort	49
4.2	Sortieren durch Auswählen II: Heapsort	52
4.2.1	Teilweise sortierte Strukturen	52
4.2.2	Heaps	54
4.2.3	Heaps als Implementation der Prioritäts-Warteschlange	54
4.2.4	Auf dem Weg zu Heapsort	58
4.2.5	Heapsort	59
4.2.6	Zusammenfassung	62
4.3	Sortieren durch Zerlegen: Quicksort	63
4.3.1	Die Idee	63
4.3.2	Partitionierung	64
4.3.3	Komplexität	68
4.3.4	Wahl des Medians	70
4.3.5	Quicksort auf kleinen Arrays	72
4.4	Vergleich	73
5	Externes Sortieren	75
5.1	Sortieren durch Mischen	75
5.1.1	Bänder	76
5.1.2	Die Idee des Mischens	76
5.1.3	Der ADT „triple-tape“	79
5.1.4	Komplexitätsbetrachtungen	83
5.2	Verbesserungen	83

6 Hashing	87
6.1 Einführung	87
6.2 Das Szenario	88
6.2.1 Beispiele für Hash-Funktionen	88
6.3 Kollisionsbehandlung	90
6.3.1 Verkettung	90
6.3.2 Sondieren	94
6.4 Schlußbemerkungen	100
7 Suchen in Baumstrukturen	101
7.1 Sortierte Binärbäume	101
7.1.1 Elementare Operationen auf sortierten Binärbäumen	103
7.1.2 Der Ordnungs-Nachfolger	105
7.1.3 Löschen aus einem sortierten Binärbaum	107
7.2 Rot/Schwarz Bäume	110
7.2.1 Definition und elementare Eigenschaften	110
7.2.2 Rotation	112
7.2.3 Einfügen in einen RS-Baum	114
7.2.4 Löschen aus einem RS-Baum	119
7.3 B-Bäume	126
7.3.1 Benutzung von B-Bäumen	127
7.3.2 Struktur von B-Bäumen	129
7.3.3 Suchen in einem B-Baum	131
7.3.4 Einfügen in einen B-Baum	132
7.3.5 Löschen aus einem B-Baum	138
7.3.6 Komplexität	148
8 Suchen in Graphen	151
8.1 Tiefensuche	151
8.2 Breitensuche	159
8.3 Kürzeste Wege I: Breitensuche	161

9 Greedy-Optimierung	163
9.1 Allgemeiner Aufbau von Greedy-Algorithmen	165
9.2 Minimalgerüste	166
9.2.1 Datenstrukturen für disjunkte Mengen	166
9.2.2 Der Kruskal-Algorithmus	168
9.3 Scheduling-Probleme	170
9.4 Datenkompression	177
9.4.1 Aufbau des Codebaums	179
9.4.2 Komplexität	185
9.4.3 Komprimierung	185
9.4.4 Expansion	186
9.5 Kürzeste Wege II: Single-Source Shortest Path	187
9.5.1 Implementation des Dijkstra-Algorithmus	187
9.5.2 Beispiel	192
9.5.3 Komplexität	193
9.6 Weitere greedy lösbare Problem	193
10 Dynamische Programmierung	195
10.1 Kürzeste Wege III: All-Pair Shortest Path	195
10.1.1 Erste Überlegungen	195
10.1.2 Der Floyd-Warshall Algorithmus	197
10.2 Der 0-1 Rucksack mit Dynamischer Programmierung	201
10.3 Parsing kontextfreier Sprachen	204
10.3.1 Die Chomsky-Normalform	204
10.3.2 Der DP-Ansatz	205
10.3.3 Beispiel	207
10.3.4 Komplexität	209
10.4 Schlußbemerkungen	210
11 Branch and Bound	211
11.1 Der 0-1 Rucksack	211
11.2 Schlußbemerkungen	217