

Algorithmen kapieren

Visuell lernen und verstehen

DAS INHALTS- VERZEICHNIS

» Hier geht's
direkt
zum Buch




Inhaltsverzeichnis



	Vorwort	13
	Geleitwort	15
	Einleitung	17
	Verwendung dieses Buchs	18
	Wer sollte dieses Buch lesen?	18
	Aufbau dieses Buchs: Überblick	18
	Konventionen und Downloads	20
	Über den Autor	20
	Über den Fachkorrektor	20
	Danksagungen	21
1	Einführung in Algorithmen	23
1.1	Einführung	23
1.1.1	Performance	24
1.1.2	Problemlösungen	24
1.2	Binäre Suche	25
1.2.1	Eine bessere Suchmethode	27
	✍ Übungen	32
1.2.2	Laufzeit	32
1.3	Landau-Notation	33
1.3.1	Die Laufzeiten von Algorithmen nehmen unterschiedlich schnell zu	33
1.3.2	Visualisierung verschiedener Laufzeiten	36

1.3.3	Die Landau-Notation beschreibt die Laufzeit im Worst Case	37
1.3.4	Typische Laufzeiten gebräuchlicher Algorithmen	38
	✎ Übungen	39
1.3.5	Das Problem des Handlungsreisenden	40
1.4	Zusammenfassung	42
2	Selectionsort	43
2.1	Die Funktionsweise des Arbeitsspeichers	44
2.2	Arrays und verkettete Listen	46
	2.2.1 Verkettete Listen	47
	2.2.2 Arrays	48
	2.2.3 Terminologie	50
	✎ Übung	50
	2.2.4 Einfügen in der Mitte einer Liste	51
	2.2.5 Löschen	53
2.3	Was wird häufiger verwendet: Arrays oder Listen?	53
	✎ Übungen	55
2.4	Selectionsort	57
2.5	Zusammenfassung	62
3	Rekursion	63
3.1	Rekursion	64
3.2	Basisfall und Rekursionsfall	67
3.3	Der Stack	68
	3.3.1 Der Aufruf-Stack	69
	✎ Übung	72
	3.3.2 Der Aufruf-Stack mit Rekursion	72
	✎ Übung	76
3.4	Zusammenfassung	76
4	Quicksort	77
4.1	Teile und herrsche	78
	✎ Übungen	85
4.2	Quicksort	86
4.3	Landau-Notation im Detail	91
	4.3.1 Mergesort und Quicksort im Vergleich	92
	4.3.2 Average Case und Worst Case im Vergleich	94
	✎ Übungen	98
4.4	Zusammenfassung	98

5	Hashtabellen	99
5.1	Hashfunktionen.	102
	✎ Übungen	106
5.2	Anwendungsfälle.	107
	5.2.1 Hashtabellen zum Nachschlagen verwenden	107
	5.2.2 Doppelte Einträge verhindern	109
	5.2.3 Hashtabellen als Cache verwenden.	111
	5.2.4 Zusammenfassung	114
	5.2.5 Kollisionen	114
5.3	Performance.	117
	5.3.1 Der Auslastungsfaktor	119
	5.3.2 Eine gute Hashfunktion.	121
	✎ Übungen	122
5.4	Zusammenfassung	122
6	Breitensuche	125
6.1	Einführung in Graphen	126
6.2	Was ist ein Graph?	128
6.3	Breitensuche	130
	6.3.1 Den kürzesten Pfad finden	132
	6.3.2 Warteschlangen	134
	✎ Übungen	135
6.4	Implementierung des Graphen	135
6.5	Implementierung des Algorithmus	138
	6.5.1 Laufzeit	143
	✎ Übungen	143
6.6	Zusammenfassung	146
7	Bäume	147
7.1	Dein erster Baum	148
	7.1.1 Dateiverzeichnisse	148
7.2	A Space Odyssey: Gefunden mit der Tiefensuche	152
	7.2.1 Eine bessere Definition von Bäumen	156
7.3	Binärbäume	156
7.4	Huffman-Codierung	157
7.5	Zusammenfassung	163
8	Balancierte Bäume.	165
8.1	Ein Balanceakt	166
	8.1.1 Schnelleres Einfügen mit Bäumen	166
8.2	Kürzere Bäume sind schneller	170

8.3	AVL-Bäume: eine Form von balancierten Bäumen	173
8.3.1	Rotationen	173
8.3.2	Woher weiß der AVL-Baum, wann es Zeit für eine Rotation ist?	176
8.4	Splay-Bäume	181
8.5	B-Bäume	183
8.5.1	Welchen Vorteil bieten B-Bäume?	184
8.6	Zusammenfassung	186
9	Der Dijkstra-Algorithmus	189
9.1	Anwendung des Dijkstra-Algorithmus	190
9.2	Terminologie	195
9.3	Eintauschen gegen ein Klavier	197
9.4	Negativ gewichtete Kanten	204
9.5	Implementierung	207
	 Übung	217
9.6	Zusammenfassung	218
10	Greedy-Algorithmen	219
10.1	Das Stundenplanproblem	219
10.2	Das Rucksackproblem	222
	 Übungen	224
10.3	Das Mengenüberdeckungsproblem	224
10.3.1	Approximationsalgorithmen	225
10.4	Zusammenfassung	231
11	Dynamische Programmierung	233
11.1	Das Rucksackproblem	233
11.1.1	Die einfache Lösung	234
11.1.2	Dynamische Programmierung	235
11.2	Häufig gestellte Fragen zum Rucksackproblem	243
11.2.1	Was geschieht beim Hinzufügen eines Gegenstands?	243
	 Übung	246
11.2.2	Was geschieht, wenn die Reihenfolge der Zeilen geändert wird?	246
11.2.3	Kann man das Gitter auch spaltenweise (statt zeilenweise) befüllen?	247
11.2.4	Was geschieht, wenn man ein leichteres Objekt hinzufügt?	247
11.2.5	Kann man Teile eines Gegenstands stehlen?	248
11.2.6	Optimierung des Reiseplans	248
11.2.7	Handhabung voneinander abhängiger Objekte	250

11.2.8	Ist es möglich, dass die Lösung mehr als zwei Teil-Rucksäcke erfordert?	251
11.2.9	Ist es möglich, dass die beste Lösung den Rucksack nicht vollständig füllt?	251
	✎ Übung	251
11.3	Der längste gemeinsame Teilstring	252
11.3.1	Erstellen des Gitters	253
11.3.2	Befüllen des Gitters	254
11.3.3	Die Lösung	255
11.3.4	Die längste gemeinsame Teilfolge	256
11.3.5	Die längste gemeinsame Teilfolge – Lösung	258
	✎ Übung	259
11.4	Zusammenfassung	259
12	k-nächste Nachbarn	261
12.1	Klassifikation von Orangen und Grapefruits	261
12.2	Entwicklung eines Empfehlungssystems	263
	12.2.1 Merkmalsextraktion	265
	✎ Übungen	269
	12.2.2 Regression	270
	12.2.3 Auswahl geeigneter Merkmale	272
	✎ Übung	273
12.3	Einführung in Machine Learning	273
	12.3.1 OCR	274
	12.3.2 Entwicklung eines Spamfilters	275
	12.3.3 Vorhersage der Entwicklung des Aktienmarkts	276
12.4	Ablauf des Trainings für ein ML-Modell im Überblick	276
12.5	Zusammenfassung	278
13	Die nächsten Schritte	281
13.1	Lineare Regression	281
13.2	Invertierte Indizes	283
13.3	Die Fourier-Transformation	284
13.4	Nebenläufige Algorithmen	284
13.5	Map/Reduce	286
	13.5.1 Warum sind verteilte Algorithmen nützlich?	286
13.6	Bloom-Filter und HyperLogLog	286
	13.6.1 Bloom-Filter	288
	13.6.2 HyperLogLog	288
13.7	HTTPS und der Diffie-Hellman-Schlüsselaustausch	289
13.8	Locality-Sensitive Hashing	293

13.9	Min-Heaps und Prioritätswarteschlangen	294
13.10	Lineare Programmierung	296
13.11	Epilog	297
A	Performance von AVL-Bäumen	299
B	NP-schwere Probleme	301
B.1	Entscheidungsprobleme	302
B.2	Das Erfüllbarkeitsproblem	303
B.3	Schwer zu lösen, schnell zu verifizieren	306
B.4	Reduktionen	308
B.5	NP-schwer	309
B.6	NP-vollständig	310
B.7	Zusammenfassung	311
C	Lösungen zu den Übungen	313
	Kapitel 1	313
	Kapitel 2	314
	Kapitel 3	317
	Kapitel 4	318
	Kapitel 5	319
	Kapitel 6	320
	Kapitel 9	322
	Kapitel 10	323
	Kapitel 11	324
	Kapitel 12	324
	Stichwortverzeichnis	327