

Auf einen Blick

| | |
|--|------------|
| Einleitung | 17 |
| Teil I: Grundbegriffe | 23 |
| Kapitel 1: Algorithmen..... | 25 |
| Kapitel 2: Qualität von Algorithmen..... | 47 |
| Kapitel 3: Daten und ihre Struktur..... | 75 |
| Teil II: Algorithmen in den Gärten der Strukturen | 111 |
| Kapitel 4: Listen: Immer einer nach dem anderen..... | 113 |
| Kapitel 5: Bäume: Immer einer über dem anderen..... | 161 |
| Kapitel 6: Graphen: Jeder mit jedem..... | 195 |
| Teil III: Probleme und ihre Lösungen | 221 |
| Kapitel 7: Sortieren..... | 223 |
| Kapitel 8: Rucksack packen..... | 253 |
| Kapitel 9: Mengen und ihre Speicherung..... | 279 |
| Kapitel 10: Verbindungen finden..... | 321 |
| Teil IV: Algorithmische Techniken | 351 |
| Kapitel 11: Probleme totschiagen..... | 353 |
| Kapitel 12: Teilen und Herrschen..... | 393 |
| Kapitel 13: Dynamisches Programmieren..... | 411 |
| Kapitel 14: Näherungslösungen..... | 437 |
| Teil V: Der Top-Ten-Teil | 465 |
| Kapitel 15: Zehn Datenabstraktionen und Datenstrukturen..... | 467 |
| Kapitel 16: Zehn Ratschläge, wenn (bevor) der kleine Frust kommt..... | 475 |
| Stichwortverzeichnis | 481 |

Inhaltsverzeichnis

| | |
|---|-----------|
| Einleitung | 17 |
| Über dieses Buch | 17 |
| Törichte Annahmen über den Leser..... | 19 |
| Wie dieses Buch aufgebaut ist | 19 |
| Symbole, die in diesem Buch verwendet werden | 20 |
| Wie es weitergeht..... | 21 |
| | |
| TEIL I | |
| GRUNDBEGRIFFE | 23 |
| | |
| Kapitel 1 | |
| Algorithmen | 25 |
| Das sind Algorithmen..... | 25 |
| Algorithmen lösen Probleme..... | 26 |
| Algorithmen basieren auf einem einfachen Maschinenmodell..... | 30 |
| Algorithmen sind bewertbar..... | 32 |
| Algorithmen agieren in Modellwelten..... | 32 |
| Algorithmen sind keine Programme..... | 33 |
| Algorithmen klar beschreiben..... | 35 |
| Sprechen Sie Pseudocode?..... | 35 |
| Mathematische Ausdrücke sind erlaubt..... | 37 |
| Algorithmen sprechen sogar Deutsch..... | 37 |
| Algorithmen sind Lösungen, keine Probleme | 38 |
| Algorithmen haben zählbare Schritte | 39 |
| Algorithmen sollten korrekt sein..... | 40 |
| Algorithmen können sich aufhängen..... | 41 |
| Das Halteproblem ist unlösbar | 42 |
| Algorithmen richtig verstehen..... | 43 |
| | |
| Kapitel 2 | |
| Qualität von Algorithmen | 47 |
| So gut sind Algorithmen | 47 |
| Wer ist der Leichteste?..... | 48 |
| Laufzeiten vergleichen..... | 50 |
| Laufzeitanalysen..... | 53 |
| Lineare Laufzeiten..... | 53 |
| Oh du großes \mathcal{O} !..... | 55 |
| Arten der Laufzeitanalyse | 57 |
| Laufzeiten konkret bestimmen..... | 59 |
| Faustregel 1: Bei Schleifen muss man multiplizieren | 59 |
| Faustregel 2: Der stärkste Summand dominiert..... | 61 |
| Vorsicht vor versteckten Kosten..... | 61 |

| | |
|--|----|
| Randomisierte Laufzeitanalyse..... | 62 |
| Das Auswahlproblem | 63 |
| QuickSelect: Ein randomisierter Algorithmus..... | 63 |
| Amortisierte Laufzeitanalyse..... | 66 |
| Ein Binärzähler an der Fassade..... | 66 |
| Ein sparsamer Stapel..... | 69 |
| Die Potenzialmethode | 71 |

Kapitel 3

| | |
|---|-----------|
| Daten und ihre Struktur..... | 75 |
| Aus Daten Strukturen bauen..... | 75 |
| Datenstrukturen: die Essenz..... | 76 |
| Datenstrukturen im Pseudocode..... | 78 |
| Algebraische Datentypen | 92 |
| Funktion folgt Struktur | 97 |
| Endrekursive und linear-rekursive Funktionen | 98 |
| Linear-rekursive Funktionen und die Akkumulatortechnik..... | 101 |
| Strukturelle Rekursion | 103 |
| Teilen und Herrschen..... | 105 |
| Strukturen durchlaufen: Iteratoren und Traversierungen..... | 106 |

TEIL II

ALGORITHMEN IN DEN GÄRTEN DER STRUKTUREN..... 111

Kapitel 4

| | |
|---|------------|
| Listen: Immer einer nach dem anderen..... | 113 |
| Listen: Datenmodell und Implementierung..... | 116 |
| Datenabstraktion: Was Listen so können sollen..... | 118 |
| Alles ist ewig und Rekursion ist gut..... | 129 |
| Listen in Funktionalistan | 131 |
| Persistente Datenstrukturen | 143 |
| Ordnung herstellen: imperativ und funktional..... | 145 |
| Nicht alles ist ewig und Form ist nicht Inhalt..... | 152 |
| Warteschlange als funktionale Datenabstraktion..... | 152 |
| Warteschlangen mit Amortisation..... | 155 |
| Rückblick: Imperative und funktionale Algorithmen | 157 |

Kapitel 5

| | |
|---|------------|
| Bäume: Immer einer über dem anderen | 161 |
| Wo ist die Kokosnuss?..... | 162 |
| Baumtraversierungen..... | 163 |
| Mit Stapeln in die Tiefe tauchen..... | 168 |
| Mit Warteschlangen in die Breite gehen..... | 173 |
| Die Kleinen nach links, die Großen nach rechts..... | 176 |
| Binäre Suchbäume | 177 |
| Verzeichnisse als Suchbäume | 179 |

| | |
|---|-----|
| Bäume verkleiden sich gerne mal..... | 181 |
| Tries | 182 |
| Prioritätswarteschlangen..... | 184 |
| Bäume als Datenmodell..... | 189 |
| Ausdrucksbäume..... | 190 |
| Mit Stapeln übersetzen und auswerten..... | 191 |

Kapitel 6

Graphen: Jeder mit jedem..... 195

| | |
|--|-----|
| Im Irrgarten der sozialen Netzwerke..... | 196 |
| Ein kurzer Blick in die Welt der Graphen..... | 198 |
| Einflussnahme als Graphenproblem..... | 202 |
| Graphen traversieren..... | 203 |
| Datenstrukturen für Graphen..... | 206 |
| Nachbarschaften dokumentieren..... | 207 |
| Daten und Probleme machen Graphen..... | 210 |
| Was nicht passt, wird passend gemacht..... | 212 |
| Erst die Schuhe, dann das Hemd – oder wie?..... | 214 |
| Topologische Sortierung, ein guter Start in den Tag..... | 214 |
| Liste folgt Funktional..... | 216 |
| Array folgt Imperativ..... | 217 |

TEIL III

PROBLEME UND IHRE LÖSUNGEN..... 221

Kapitel 7

Sortieren..... 223

| | |
|---|-----|
| Alles in Ordnung..... | 223 |
| Das Sortierproblem..... | 224 |
| SelectionSort: So lange wählen, bis es passt..... | 227 |
| Laufzeit von SelectionSort..... | 228 |
| MergeSort: Geteiltes Leid ist halb sortiert..... | 229 |
| Sortierte Teilarrays verschmelzen mit Merge..... | 230 |
| Teilen und Herrschen..... | 232 |
| Laufzeit von MergeSort..... | 232 |
| QuickSort: Quick and Easy..... | 234 |
| Partition teilt das Array auf..... | 234 |
| Sortieren mit QuickSort..... | 235 |
| Worst-Case-Laufzeit von QuickSort..... | 236 |
| Randomisierung..... | 237 |
| HeapSort: Ein Haufen Arbeit..... | 237 |
| Die Datenstruktur Heap..... | 238 |
| Der Heap als Priority Queue..... | 239 |
| Besser sortieren mit dem Heap..... | 240 |
| Die maximale Sortiergeschwindigkeit..... | 241 |

12 Inhaltsverzeichnis

| | |
|---|-----|
| Sortieren in Linearzeit..... | 244 |
| CountingSort: Sortieren durch Zählen | 244 |
| Sortieren nach Ziffern..... | 245 |
| Stabile Sortierverfahren..... | 247 |
| RadixSort: Mehrfach ziffernweise sortieren..... | 248 |
| Weitere Sortieralgorithmen..... | 249 |
| BubbleSort: Nachbarn vertauschen..... | 249 |
| Gnomesort: Immer hin und her..... | 250 |
| InsertionSort: Spielkarten dazwischen schieben..... | 251 |

Kapitel 8

Rucksack packen 253

| | |
|--|-----|
| Wie man einen Koffer packt..... | 253 |
| Das Rucksackproblem | 253 |
| Das Wichtigste zuerst einpacken..... | 255 |
| Alles ausprobieren | 257 |
| Suchen im Entscheidungsbaum..... | 258 |
| Den Suchraum begrenzen..... | 261 |
| Probleme langsam wachsen lassen..... | 264 |
| Wachsende Probleme klug speichern..... | 267 |
| Sich dem Optimum annähern | 270 |
| Lineare Optimierung..... | 274 |
| Optimierung von Produktionsmengen..... | 274 |
| Ein System von Ungleichungen..... | 275 |
| Ziel: Gewinnmaximierung | 275 |
| Ganzzahlige lineare Optimierung..... | 276 |
| Das Rucksackproblem als ILP | 277 |

Kapitel 9

Mengen und ihre Speicherung 279

| | |
|---|-----|
| Ich bin eine Menge | 281 |
| Imperative Datenabstraktion für Mengen | 283 |
| Funktionale Datenabstraktion für Mengen..... | 285 |
| Gut gehackt ist schnell gefunden..... | 290 |
| Hashfunktionen..... | 292 |
| Hashtabellen..... | 293 |
| Garantiert gut gehackt..... | 298 |
| Derselbe ist nicht immer der Gleiche..... | 300 |
| Viel ist oft eine Menge..... | 304 |
| Wer Ordnung hält, ist nur zu faul zum Suchen..... | 306 |
| Bäume balancieren | 308 |
| Rot-Schwarz-Bäume | 311 |

Kapitel 10

Verbindungen finden 321

| | |
|---|-----|
| Kürzeste Pfade..... | 322 |
| Alle kürzesten Pfade von einem Start aus..... | 322 |
| Vom Vertrauten ins Unbekannte..... | 325 |

| | |
|---|-----|
| Kürzester Pfad zu allen Knoten..... | 328 |
| Dijkstras Algorithmus..... | 330 |
| Datenstrukturen für Dijkstras Algorithmus..... | 333 |
| Verbundenes aufspüren..... | 334 |
| Verbundene Komponenten identifizieren..... | 335 |
| Datenstrukturen bei der Berechnung verbundener Komponenten..... | 338 |
| Disjunkte Mengen als Datenstruktur..... | 340 |
| Laufzeiten..... | 344 |
| Spann mir einen Graphen auf..... | 345 |
| Minimaler Spannbaum..... | 346 |
| Kruskals Algorithmus..... | 347 |

TEIL IV
ALGORITHMISCHE TECHNIKEN 351

Kapitel 11
Probleme totschiagen 353

| | |
|---|-----|
| Erschöpfende Suche..... | 354 |
| Die üblichen Verdächtigen: Kombinatorische Objekte..... | 355 |
| Konzentrierte oder weit ausschweifende Suche..... | 358 |
| Die erschöpfende Suche nach acht friedlichen Damen..... | 362 |
| Iterative und rekursive Erzeugung des Suchraums..... | 364 |
| Schleifen rekursiv erzeugen..... | 364 |
| Einen baumartigen Suchraum rekursiv erzeugen..... | 366 |
| Backtracking..... | 369 |
| Kandidaten nicht stückweise bewertbar: kein Backtracking..... | 371 |
| Backtracking als Suche im Zustandsraum..... | 373 |
| Verzweigen und Begrenzen..... | 375 |
| Erschöpfende und Backtracking-Suche im Irrgarten..... | 375 |
| Optimierungen und Bewertungsfunktionen..... | 377 |
| Komplexitätsklassen: Schwere Probleme führen zu anstrengender Arbeit..... | 380 |
| Schwer ist, was den Besten schwerfällt..... | 380 |
| Ein Labyrinth der Kameras..... | 382 |
| Das nichtdeterministische Orakel..... | 383 |
| Schwer, schwerer, NP-schwer..... | 385 |
| Wie man mit schweren Problemen umgeht..... | 387 |
| NP-schwer \neq hoffnungslos..... | 387 |
| Gute Ideen sind kein Hexenwerk..... | 390 |

Kapitel 12
Teilen und Herrschen 393

| | |
|--|-----|
| Aufgaben auf Mitarbeiter abwälzen..... | 393 |
| Das Einwohnermeldeamt von Bürokrazien..... | 393 |
| Das Prinzip Teilen und Herrschen..... | 395 |
| Laufzeiten bei Teilen und Herrschen..... | 396 |
| Das Mastertheorem..... | 397 |
| Fall 1: Der Chef arbeitet mehr..... | 398 |
| Fall 2: Der Chef arbeitet gleich viel..... | 399 |

| | |
|--|-----|
| Fall 3: Der Chef arbeitet weniger | 400 |
| Gibt es noch weitere Fälle? | 401 |
| So bestimmt man, welcher Fall vorliegt | 401 |
| Binärsuche | 403 |
| Der Suchbaum in einfach | 403 |
| Grenzen des Suchbereichs | 405 |
| Weitere Beispiele für Teilen und Herrschen | 407 |
| Sortieren | 407 |
| Matrizen multiplizieren | 408 |
| Minimaler Punktabstand | 409 |

Kapitel 13

| | |
|---|------------|
| Dynamisches Programmieren | 411 |
| Ein profitabler Bauauftrag | 411 |
| Das Maximale-Teilsumme-Problem | 412 |
| Gier hilft nicht | 412 |
| Rohe Gewalt hilft eher | 413 |
| Inkrementelle Gewalt ist weniger roh | 413 |
| Ein Stück abschneiden und Herrschen | 414 |
| Zwischenergebnisse merken | 416 |
| Den Algorithmus vom Kopf auf die Füße stellen | 418 |
| Der ultimative Maximale-Teilsumme-Algorithmus | 418 |
| Probleme wachsen lassen | 419 |
| Das Prinzip des dynamischen Programmierens | 419 |
| Beispiel 1: Minimum | 420 |
| Beispiel 2: Fibonacci-Zahlen | 421 |
| Beispiel 3: Rucksack packen | 424 |
| Vergleich von Texten | 424 |
| Die Editierdistanz | 425 |
| Strings alignieren | 426 |
| Arbeitsteilung auf der Alignmentbaustelle | 427 |
| Optimale Alignments mit dynamischem Programmieren | 428 |
| Der Weg zum Optimum | 431 |
| Entscheidungen merken | 431 |
| Den Pfad zurückfinden | 433 |

Kapitel 14

| | |
|--|------------|
| Näherungslösungen | 437 |
| Heuristiken | 437 |
| Interpolationssuche | 438 |
| Heuristisches Verzweigen und Begrenzen | 441 |
| Der A*-Algorithmus | 443 |
| Approximation | 446 |
| TSP: Die kürzeste Rundreise | 446 |
| Gierige Heuristik | 447 |
| Lokale Suche | 449 |
| Approximation ohne Heuristik | 450 |

| | |
|--|------------|
| Gier..... | 453 |
| Das Wechselgeldproblem..... | 455 |
| Das Problem der Mengenüberdeckung..... | 458 |
| Gier in Perfektion..... | 461 |
| Huffman-Codierung..... | 461 |
| TEIL V | |
| DER TOP-TEN-TEIL..... | 465 |
| Kapitel 15 | |
| Zehn Datenabstraktionen und Datenstrukturen..... | 467 |
| Stapel..... | 468 |
| Warteschlange..... | 469 |
| Prioritätswarteschlange..... | 469 |
| Liste..... | 470 |
| Array..... | 471 |
| Menge..... | 471 |
| Verzeichnis..... | 472 |
| Relation..... | 472 |
| Graph..... | 473 |
| Baum..... | 474 |
| Kapitel 16 | |
| Zehn Ratschläge, wenn (bevor) der kleine Frust kommt..... | 475 |
| Rekursion ist deine Freundin..... | 475 |
| Mathematik ist einfach..... | 476 |
| Pseudocode ist verstehbar..... | 477 |
| Abstraktion ist gut..... | 477 |
| Sei auch mal funktional..... | 478 |
| Ein Bild sagt mehr als 1000 Worte..... | 478 |
| Vieles ist solides Handwerk..... | 479 |
| Es geht auch um Kreativität..... | 479 |
| Unterscheide Datenmodell und Datenstruktur..... | 480 |
| Was schwierig aussieht, ist oft auch schwierig..... | 480 |
| Stichwortverzeichnis..... | 481 |