

# Inhalt

Einleitung	11
1 Vorbemerkungen: Stellenwert des Themas, Hintergründe, Begründungs- und Bedeutungszusammenhänge, Ziele	18
2 Begriffsbestimmungen	24
2.1 Zum Begriff des Algorithmus	24
2.2 Zum Begriff der Informatik	31
3 Historische Bezüge	35
3.1 Ein Exkurs zur Geschichte der Algorithmik und Informatik	35
3.1.1 Zur Entwicklung der schriftlichen Rechenverfahren	48
3.1.2 Algebraisierung – die Idee der „ars magna“	51
3.1.3 Zur Geschichte der Rechenmaschinen	52
3.2 Vier klassische Algorithmen	56
3.2.1 Das sumerisch-babylonische Wurzelziehen bzw. Heron-Verfahren	57
3.2.2 Der Euklidische Algorithmus	62
3.2.3 Das Sieb des Eratosthenes	72
3.2.4 Die Approximation von $\pi$ nach Archimedes	78
3.3 Algorithmisches Definieren und Beweisen	88
3.3.1 Die Unendlichkeit der Primzahlmenge	94
3.3.2 Inkommensurabilitätsbeweise	95
4 Fundamentale heuristische Strategien des algorithmischen Problemlösens	99
4.1 Elementare Methoden	99
4.1.1 Die Methode der rohen Gewalt (brute force method)	100
4.1.2 Die gierige Strategie (greedy strategy)	102
• Ägyptische Bruchrechnung	102
• Arbeitsplanung (job scheduling)	107
• Konstruktion eines minimalen Gerüsts (minimal spanning tree)	109
4.2 Methoden, die sich stark am Einsatz von Computern orientieren	114
4.2.1 Modularität	114
4.2.2 Rekursion	118
• Das Turm-von-Hanoi Spiel	121

4.2.3	Das Prinzip „Teile und Herrsche“ (divide et impera, divide and conquer)	123
	• Quicksort	124
4.3	Methoden, die im Zusammenhang mit der Bearbeitung von Bäumen und Graphen zur Anwendung kommen	127
4.3.1	Systematisches Durchlaufen von Baumstrukturen	128
	• Algorithmus Tiefensuche	132
	• Algorithmus Breitensuche	135
	• Das Rucksack-Problem (knapsack problem)	136
4.3.2	Backtracking	141
	• Das Damenproblem	141
4.4	Die gezielte mathematische Analyse	145
	• Das NIM-Spiel	146
4.5	Probabilistische Verfahren, Modellbildung und Simulation	148
	• Das Sammlerproblem	152
	• Das Ziegenproblem	154
4.6	Parallelität	158
5	Effizienz von Algorithmen	159
5.1	Iteration und Rekursion unter dem Gesichtspunkt der Effizienz	160
5.2	Kognitive Effizienz	165
5.3	Das Prinzip von „Teile und Herrsche“ unter dem Aspekt der Effizienz	166
	• Schnelles Potenzieren	167
5.4	Das Horner-Schema	172
	• Erster Exkurs: Stellenwertsysteme	175
	• Zweiter Exkurs: Die Einkommensteuer	177
5.5	Die Zeitkomplexität des Euklidischen Algorithmus	178
5.6	Einige wichtige Funktionstypen zur Beschreibung der Effizienz von Algorithmen	180
5.7	Algorithmisch aufwendige Probleme	183
	• Das Königsberger Brückenproblem	184
	• Eulersche und Hamiltonsche Wege	184
	• Das Traveling Salesman Problem	189
	• Entscheidungsprobleme und Optimierungsprobleme	189
	• Die Komplexitätsklassen P, NP und NP-vollständig	190

6	Korrektheit von Computerergebnissen	
	Korrektheit von Algorithmen	195
6.1	Fehler in der Arithmetik von Computern	197
6.2	Partielle und totale Korrektheit von Algorithmen	205
6.3	Formale Methoden	206
7	Grenzen des Computers / Grenzen der Algorithmisierbarkeit	210
7.1	Entwicklung der wissenschaftstheoretischen Grundideen	213
7.2	Formalisierung des Algorithmus-Begriffs / der Begriff der Berechenbarkeit	225
7.3	Einige konkrete, algorithmisch nicht lösbare Probleme	226
	• Das Halteproblem	227
8	Programmierung	230
8.1	Zum Verhältnis von „Maschinensprachen“ und „höheren“ Programmiersprachen	231
8.2	Wie werden die in einer höheren Programmiersprache geschriebenen Programme verarbeitet?	236
8.3	Paradigmen des Programmierens Programmiersprachen-Familien	238
8.4	Die wichtigsten Kontrollstrukturen in strukturierten Programmiersprachen	241
	• Die Anweisungsfolge (Sequenz)	242
	• Die Fallunterscheidung (Auswahl, Verzweigung)	242
	• Die Wiederholung (Schleife)	243
	• Kontrollstrukturen und Modularität	244
	• Der Sprungbefehl	244
	• Strukturiertes Programmieren	245
	• Flussdiagramme	245
8.5	Die wichtigsten Datenstrukturen	248
	• Numerische Datentypen	251
	• Der Datentyp des Feldes	252
	• Der Datentyp des Verbunds	252
	• Der Datentyp der Liste	253
8.6	Modulares Programmieren mit Prozeduren und Funktionen	255
	• Prozeduren	256
	• Funktionen	257

8.7	Diskussion einiger konkreter Programmiersprachen	261
8.7.1	Die Familie der ALGOL-ähnlichen Programmierersprachen	261
	• Pascal	263
8.7.2	Programmierersprachen aus dem Bereich der „Künstlichen Intelligenz“	265
	• Lisp	265
	• Logo	267
	• Scheme	268
	• Prolog	270
	• Computeralgebra Systeme	274
8.7.3	Das Phänomen der Interaktivität	274
	• BASIC	275
8.8	Programmierungsumgebungen, Betriebssysteme, Benutzerschnittstellen und Anwendersysteme	277
9	Evolutionäre Algorithmen und neuronale Netze	280
9.1	Evolutionäre Algorithmen	281
	• Die Methode der evolutionären Algorithmen – erläutert am Traveling Salesman Problem	287
9.2	Neuronale Netze	297
9.2.1	Backpropagation-Netze	309
9.2.2	Rückgekoppelte Netze (Hopfield-Netze)	315
	• Mustererkennung mit rückgekoppelten Netzen	318
9.2.3	Selbstorganisierende Netze (Kohonen-Netze)	323
	• Selbstorganisierende Netze am Beispiel des Traveling Salesman Problems	328
	Bildquellen	334
	Literaturverzeichnis	335
	Index	343