

Inhalt

Einleitung	7
1 Vorbemerkungen: Stellenwert des Themas, Hintergründe, Begründungs- und Bedeutungszusammenhänge, Ziele	14
2 Begriffsbestimmungen	19
2.1 Zum Begriff des Algorithmus	19
2.2 Zum Begriff der Informatik	26
3 Historische Bezüge	30
3.1 Exkurs zur Geschichte der Algorithmik und Informatik	30
• Zur Entwicklung der schriftlichen Rechenverfahren	44
• Algebraisierung – die Idee der „ars magna“	47
• Zur Geschichte der Rechenmaschinen	48
3.2 Vier klassische Algorithmen	52
3.2.1 Das sumerisch-babylonische Wurzelziehen bzw. Heron-Verfahren	54
3.2.2 Der Euklidische Algorithmus	60
3.2.3 Das Sieb des Eratosthenes	71
3.2.4 Die Approximation von π (nach Archimedes)	79
3.3 Algorithmisches Definieren und Beweisen	89
3.3.1 Die Unendlichkeit der Primzahlmenge	96
3.3.2 Inkommensurabilitätsbeweise	97
4 Heuristische Strategien des algorithmischen Problemlösens	101
4.1 Elementarst-Methoden	101
4.1.1 Die Methode der rohen Gewalt (brute force method)	102
4.1.2 Die gierige Strategie (greedy strategy)	105
• Ägyptische Bruchrechnung	105
• Arbeitsplanung (job scheduling)	110
• Konstruktion eines minimalen Gerüsts (minimal spanning tree)	112
4.2 Methoden, die sich stark am Einsatz von Computern orientieren	117
4.2.1 Modularität	117
4.2.2 Rekursion	121

• Das Turm-von-Hanoi-Spiel	124
4.2.3 Das Prinzip „Teile und Herrsche“ (divide and conquer)	127
• Quicksort	127
4.3 Methoden, die im Zusammenhang mit der Bearbeitung von Bäumen und Graphen zur Anwendung kommen	131
4.3.1 Systematisches Durchlaufen von Baumstrukturen	132
• Algorithmus Tiefensuche	136
• Algorithmus Breitensuche	139
• Das Rucksack-Problem	140
4.3.2 Backtracking	145
• Das Damenproblem	145
4.4 Die gezielte mathematische Analyse	152
• Das NIM-Spiel	152
4.5 Probabilistische Verfahren / Simulation	155
• Das Sammlerproblem	159
• Das Ziegenproblem	161
4.6 Parallelität	165
5 Effizienz von Algorithmen	167
5.1 Iteration und Rekursion unter dem Gesichtspunkt der Effizienz	168
5.2 Kognitive Effizienz	174
5.3 Das Prinzip von „Teile und Herrsche“ unter dem Aspekt der Effizienz	176
• Schnelles Potenzieren	176
5.4 Das Horner-Schema	182
• Stellenwertsysteme	185
• Die Einkommensteuer	186
5.5 Die Zeitkomplexität des Euklidischen Algorithmus	187
5.6 Einige wichtige Funktionstypen zur Beschreibung der Effizienz von Algorithmen	189
5.7 Algorithmisch „harte“ Probleme	192
• Das Königsberger Brückenproblem	193
• Eulersche und Hamiltonsche Wege	194
• Das Traveling Salesman Problem	199
• Die Komplexitätsklassen P, NP und NP-vollständig	200

6	Korrektheit von Computerergebnissen	
	Korrektheit von Algorithmen	205
6.1	Fehler in der Arithmetik von Computern	208
6.2	Partielle und totale Korrektheit von Algorithmen	216
6.3	Formale Methoden	217
	• Schleifeninvarianten	219
7	Grenzen des Computers / Grenzen der Algorithmisierbarkeit	222
7.1	Entwicklung der wissenschaftstheoretischen Grundideen	225
7.2	Formalisierung des Algorithmus-Begriffs / der Begriff der Berechenbarkeit	237
7.3	Einige konkrete, algorithmisch nicht lösbare Probleme	238
	• Das Halteproblem	239
8	Programmierung	242
8.1	Zum Verhältnis von „Maschinensprachen“ und „höheren“ Programmiersprachen	243
8.2	Wie werden die in einer höheren Programmiersprache geschriebenen Programme verarbeitet?	249
8.3	Paradigmen des Programmierens Programmiersprachen-Familien	251
8.4	Die wichtigsten Kontrollstrukturen in strukturierten Programmiersprachen	254
	• Die Anweisungsfolge (Sequenz)	254
	• Die Fallunterscheidung (Auswahl, Verzweigung)	255
	• Die Wiederholung („Schleife“)	257
	• Kontrollstrukturen und Modularität	257
	• Der Sprungbefehl	258
	• Strukturiertes Programmieren	259
	• Flußdiagramme	259
8.5	Die wichtigsten Datenstrukturen	262
	• Numerische Datentypen	266
	• Der Datentypen „Feld“	267
	• Der Datentyp „Verbund“	268
	• Der Datentyp „Liste“	270
8.6	Modulares Programmieren mit Prozeduren und Funktionen	271
	• Prozeduren	273
	• Funktionen	275

8.7	Diskussion einiger konkreter Programmiersprachen	279
8.7.1	Die Familie der ALOGOL-ähnlichen Programmiersprachen	280
	• Pascal	282
8.7.2	Programmiersprachen aus dem Bereich der „Künstlichen Intelligenz“	285
	• Lisp	285
	• Logo	287
	• Scheme	288
	• Prolog	290
	• Computeralgebra Systeme	294
8.7.3	Das Phänomen der Interaktivität	294
	• BASIC	296
8.8	Programmierungsumgebungen, Betriebssysteme, Benutzerschnittstellen und Anwendersysteme	298
9	Evolutionäre Algorithmen und neuronale Netze	301
9.1	Evolutionäre Algorithmen	302
	• Die Methode der evolutionären Algorithmen – erläutert am Traveling Salesman Problem	308
	• Genetische Algorithmen, Evolutionary Programming und Evolutionsstrategien	319
9.2	Neuronale Netze	321
9.2.1	Backpropagation-Netze	332
9.2.2	Rückgekoppelte Netze (Hopfield-Netze)	339
	• Mustererkennung mit rückgekoppelten Netzen	342
9.2.3	Selbstorganisierende Netze (Kohonen-Netze)	348
	• Selbstorganisierende Netze am Beispiel des Traveling Salesman Problems	353
	Bildquellen	359
	Literaturverzeichnis	360
	Index	367