

# COMPUTERS AND INTRACTABILITY

## A Guide to the Theory of NP-Completeness

Michael R. Garey / David S. Johnson

BELL LABORATORIES  
MURRAY HILL, NEW JERSEY



W. H. FREEMAN AND COMPANY  
New York

---

# Contents

<b>Preface</b> . . . . .	ix
<b>1 Computers, Complexity, and Intractability</b> . . . . .	1
1.1 Introduction . . . . .	1
1.2 Problems, Algorithms, and Complexity . . . . .	4
1.3 Polynomial Time Algorithms and Intractable Problems . . . . .	6
1.4 Provably Intractable Problems . . . . .	11
1.5 NP-Complete Problems . . . . .	13
1.6 An Outline of the Book . . . . .	14
<b>2 The Theory of NP-Completeness</b> . . . . .	17
2.1 Decision Problems, Languages, and Encoding Schemes . . . . .	18
2.2 Deterministic Turing Machines and the Class P . . . . .	23
2.3 Nondeterministic Computation and the Class NP . . . . .	27
2.4 The Relationship Between P and NP . . . . .	32
2.5 Polynomial Transformations and NP-Completeness . . . . .	34
2.6 Cook's Theorem . . . . .	38
<b>3 Proving NP-Completeness Results</b> . . . . .	45
3.1 Six Basic NP-Complete Problems . . . . .	46
3.1.1 3-SATISFIABILITY . . . . .	48
3.1.2 3-DIMENSIONAL MATCHING . . . . .	50
3.1.3 VERTEX COVER and CLIQUE . . . . .	53
3.1.4 HAMILTONIAN CIRCUIT . . . . .	56
3.1.5 PARTITION . . . . .	60
3.2 Some Techniques for Proving NP-Completeness . . . . .	63
3.2.1 Restriction . . . . .	63
3.2.2 Local Replacement . . . . .	66
3.2.3 Component Design . . . . .	72
3.3 Some Suggested Exercises . . . . .	74

<b>4</b>	<b>Using NP-Completeness to Analyze Problems</b>	77
4.1	Analyzing Subproblems	80
4.2	Number Problems and Strong NP-Completeness	90
4.2.1	Some Additional Definitions	92
4.2.2	Proving Strong NP-Completeness Results	95
4.3	Time Complexity as a Function of Natural Parameters	106
<b>5</b>	<b>NP-Hardness</b>	109
5.1	Turing Reducibility and NP-Hard Problems	109
5.2	A Terminological History	118
<b>6</b>	<b>Coping with NP-Complete Problems</b>	121
6.1	Performance Guarantees for Approximation Algorithms	123
6.2	Applying NP-Completeness to Approximation Problems	137
6.3	Performance Guarantees and Behavior "In Practice"	148
<b>7</b>	<b>Beyond NP-Completeness</b>	153
7.1	The Structure of NP	154
7.2	The Polynomial Hierarchy	161
7.3	The Complexity of Enumeration Problems	167
7.4	Polynomial Space Completeness	170
7.5	Logarithmic Space	177
7.6	Proofs of Intractability and P vs. NP	181
	<b>Appendix: A List of NP-Complete Problems</b>	187
A1	Graph Theory	190
A1.1	Covering and Partitioning	190
A1.2	Subgraphs and Supergraphs	194
A1.3	Vertex Ordering	199
A1.4	Iso- and Other Morphisms	202
A1.5	Miscellaneous	203
A2	Network Design	206
A2.1	Spanning Trees	206
A2.2	Cuts and Connectivity	209
A2.3	Routing Problems	211
A2.4	Flow Problems	214
A2.5	Miscellaneous	218
A3	Sets and Partitions	221
A3.1	Covering, Hitting, and Splitting	221
A3.2	Weighted Set Problems	223
A4	Storage and Retrieval	226
A4.1	Data Storage	226
A4.2	Compression and Representation	228
A4.3	Database Problems	232

A5	Sequencing and Scheduling . . . . .	236
	A5.1 Sequencing on One Processor . . . . .	236
	A5.2 Multiprocessor Scheduling . . . . .	238
	A5.3 Shop Scheduling . . . . .	241
	A5.4 Miscellaneous . . . . .	243
A6	Mathematical Programming . . . . .	245
A7	Algebra and Number Theory . . . . .	249
	A7.1 Divisibility Problems . . . . .	249
	A7.2 Solvability of Equations . . . . .	250
	A7.3 Miscellaneous . . . . .	252
A8	Games and Puzzles . . . . .	254
A9	Logic . . . . .	259
	A9.1 Propositional Logic . . . . .	259
	A9.2 Miscellaneous . . . . .	261
A10	Automata and Language Theory . . . . .	265
	A10.1 Automata Theory . . . . .	265
	A10.2 Formal Languages . . . . .	267
A11	Program Optimization . . . . .	272
	A11.1 Code Generation . . . . .	272
	A11.2 Programs and Schemes . . . . .	275
A12	Miscellaneous . . . . .	279
A13	Open Problems . . . . .	285
 <b>Symbol Index</b> . . . . .		<b>289</b>
 <b>Reference and Author Index</b> . . . . .		<b>291</b>
 <b>Subject Index</b> . . . . .		<b>327</b>
 <b>Update for the Current Printing</b> . . . . .		<b>339</b>