

# 目次

詳細な解説付き問題の一覧	xiii
詳細な解説付きアルゴリズムの一覧	xv
序文	xvii
<b>第 1 章 はじめに：いくつかの代表的問題</b>	<b>1</b>
1.1 最初の問題：安定マッチング	1
1.2 五つの代表的な問題	11
解答付き演習問題	18
演習問題	22
ノートと発展文献	27
<b>第 2 章 アルゴリズム解析の基礎事項</b>	<b>29</b>
2.1 計算容易性	29
2.2 増加の漸近的オーダー	34
2.3 リストと配列による安定マッチングアルゴリズムの実装	41
2.4 よく現れる計算時間の復習	45
2.5 より複雑なデータ構造：優先順位付きキュー	54
解答付き演習問題	61
演習問題	63
ノートと発展文献	66
<b>第 3 章 グラフ</b>	<b>67</b>
3.1 基本的定義と応用	67
3.2 グラフの連結性とグラフ走査	72
3.3 キューとスタックを用いたグラフ走査	79
3.4 二部グラフ性の判定：幅優先探索の応用	85
3.5 有向グラフの連結性	88

3.6	有向無閉路グラフとトポロジカル順序付け	90
	解答付き演習問題	94
	演習問題	97
	ノートと発展文献	102
<b>第4章</b>	<b>グリーディアルゴリズム</b>	<b>103</b>
4.1	区間スケジューリング：グリーディアルゴリズムの先進性	104
4.2	遅延最小化スケジューリング：交換議論	112
4.3	最適キャッシング：より複雑な交換議論	118
4.4	グラフにおける最短パス	124
4.5	最小全点木問題	129
4.6	Kruskal のアルゴリズムの実装：Union-Find データ構造	137
4.7	クラスタリング	143
4.8	Huffman 符号とデータ圧縮	146
* 4.9	最小コスト有向木：多フェーズグリーディアルゴリズム	160
	解答付き演習問題	166
	演習問題	171
	ノートと発展文献	185
<b>第5章</b>	<b>分割統治法</b>	<b>187</b>
5.1	最初の漸化式：マージソートアルゴリズム	188
5.2	さらなる漸化式	192
5.3	反転数の数え上げ	197
5.4	最近点对の発見	201
5.5	整数の積の計算	207
5.6	畳込みと高速フーリエ変換	209
	解答付き演習問題	216
	演習問題	219
	ノートと発展文献	222
<b>第6章</b>	<b>動的計画法</b>	<b>223</b>
6.1	重み付き区間スケジューリング：再帰的手続き	224
6.2	動的計画法の原理：部分問題上での記憶化と反復	229
6.3	区分最小二乗法：多肢選択	232
6.4	部分集合和とナップサック：変数の追加	236
6.5	RNA 二次構造：区間上での動的計画法	242
6.6	系列アライメント	246
6.7	分割統治法による線形の領域の系列アライメント	253
6.8	グラフの最短パス	258
6.9	最短パスと距離ベクトルプロトコル	265
* 6.10	グラフの負閉路	269

目 次	ix
解答付き演習問題 .....	274
演習問題 .....	278
ノートと発展文献 .....	298
<b>第 7 章 ネットワークフロー</b> .....	<b>301</b>
7.1 最大フロー問題と Ford-Fulkerson アルゴリズム .....	302
7.2 ネットワークの最大フローと最小カット .....	310
7.3 良い増加パスの選択 .....	315
* 7.4 プリフロープッシュ最大フローアルゴリズム .....	320
7.5 最初の応用：二部グラフのマッチング問題 .....	330
7.6 有向グラフと無向グラフにおける素パス .....	336
7.7 最大フロー問題の拡張版 .....	340
7.8 市場調査デザイン .....	346
7.9 航空スケジューリング .....	348
7.10 画像分割 .....	352
7.11 プロジェクト選択 .....	356
7.12 野球ペナントレースの優勝の可能性の消去 .....	359
* 7.13 発展：辺にコストの付随するマッチング問題 .....	364
解答付き演習問題 .....	371
演習問題 .....	374
ノートと発展文献 .....	404
<b>第 8 章 <math>\mathcal{NP}</math> と計算困難性</b> .....	<b>407</b>
8.1 多項式時間リダクション .....	408
8.2 “ガジェット”を用いたリダクション：充足可能性問題 .....	414
8.3 効率的な証明と $\mathcal{NP}$ の定義 .....	419
8.4 NP-完全問題 .....	422
8.5 系列化問題 .....	429
8.6 分割問題 .....	436
8.7 グラフ彩色問題 .....	441
8.8 数値問題 .....	446
8.9 $\text{co-NP}$ と $\mathcal{NP}$ の非対称性 .....	450
8.10 計算困難な問題の部分的な分類 .....	453
解答付き演習問題 .....	456
演習問題 .....	460
ノートと発展文献 .....	482
<b>第 9 章 PSPACE：クラス <math>\mathcal{NP}</math> を超える問題のクラス</b> .....	<b>485</b>
9.1 クラス PSPACE .....	485
9.2 PSPACE に属する困難な問題 .....	487
9.3 限量化問題とゲームの多項式領域解法 .....	489

9.4	計画問題の多項式領域解法	491
9.5	PSPACE-完全性の証明	496
	解答付き演習問題	499
	演習問題	501
	ノートと発展文献	503
<b>第 10 章</b>	<b>計算容易性の拡大</b>	<b>505</b>
10.1	小さい点カバリの発見	506
10.2	木における NP-困難問題の解法	509
10.3	円弧集合の彩色	513
* 10.4	グラフの木分解	522
* 10.5	木分解の構成	532
	解答付き演習問題	538
	演習問題	541
	ノートと発展文献	545
<b>第 11 章</b>	<b>近似アルゴリズム</b>	<b>547</b>
11.1	グリーディアルゴリズムと最適解に対する近似解の上界：負荷均等化問題への適用	548
11.2	センター選択問題	553
11.3	集合カバリ：一般的なグリーディヒューリスティック	559
11.4	価格付け法：点カバリへの適用	564
11.5	価格付け法による最大化：素パス問題	569
11.6	線形計画法とラウンディング：点カバリへの適用	575
* 11.7	負荷均等化問題（再考）：より高度な LP の応用	581
11.8	究極の近似保証：ナップサック問題	587
	解答付き演習問題	592
	演習問題	593
	ノートと発展文献	600
<b>第 12 章</b>	<b>局所探索</b>	<b>603</b>
12.1	最適化問題の景観図	604
12.2	Metropolis アルゴリズムとシミュレーテッドアニーリング	608
12.3	Hopfield ニューラルネットワークへの局所探索の応用	612
12.4	局所探索による最大カットの近似	616
12.5	近傍関係の選択	619
* 12.6	局所探索による分類	621
12.7	最善反応行動と Nash 均衡	629
	解答付き演習問題	638
	演習問題	640
	ノートと発展文献	643

---

<b>第 13 章 乱択アルゴリズム</b>	<b>645</b>
13.1 第一の応用：競合の解消	646
13.2 大域的最小カットの発見	651
13.3 ランダム変数と期待値	656
13.4 MAX 3-SAT に対する乱択近似アルゴリズム	661
13.5 乱択分割統治法：中央値発見とクイックソート	664
13.6 ハッシング：辞書の乱択実装	670
13.7 最近点对を求める乱択アプローチ	677
13.8 乱択キャッシング	684
13.9 Chernoff 限界	692
13.10 負荷均等化	693
13.11 パケットルーティング	696
13.12 背景：いくつかの基本的な確率の定義	701
解答付き演習問題	708
演習問題	713
ノートと発展文献	722
<b>第 14 章 永遠に動作するアルゴリズム</b>	<b>725</b>
参考文献	735
人名索引	743
問題索引	747
アルゴリズム索引	751
用語和（英）索引	755
用語英（和）索引	779



# 詳細な解説付き問題の一覧

問題 1.1	安定マッチング (Stable Matching)	1
問題 1.2	区間スケジューリング (Interval Scheduling)	12
問題 1.3	重み付き区間スケジューリング (Weighted Interval Scheduling)	13
問題 1.4	二部グラフのマッチング (Bipartite Matching)	14
問題 1.5	独立集合 (Independent Set)	15
問題 1.6	競争的施設配置 (Competitive Facility Location)	17
問題 2.1	優先順位付きキューの実装 (Implementing a Priority Queue)	54
問題 3.1	二部グラフ性の判定 (Bipartiteness)	86
問題 3.2	トポロジカル順序付け (Topological Ordering)	91
問題 4.1	区間分割 (Interval Partitioning)	109
問題 4.2	遅延最小化スケジューリング (Scheduling to Minimize Lateness)	113
問題 4.3	キャッシュ管理 (Cache Maintenance)	118
問題 4.4	最短パス (Shortest Path)	124
問題 4.5	最小全点木 (Minimum Spanning Tree)	129
問題 4.6	Union-Find データ構造 (Union-Find Data Structure) の実装	138
問題 4.7	クラスタリング (Clustering)	144
問題 4.8	符号とデータ圧縮 (Codes and Data Compression)	147
問題 4.9	最小コスト有向木 (Minimum-Cost Arborescence)	161
問題 5.1	反転数の数え上げ (Counting Inversions)	197
問題 5.2	最近点对 (Closest Pair of Points)	201
問題 5.3	整数の積の計算 (Integer Multiplication)	207
問題 5.4	畳込み (Convolution)	209
問題 6.1	区分最小二乗 (Segmented Least Squares)	232
問題 6.2	部分集合和 (Subset Sum)	237
問題 6.3	RNA 二次構造 (RNA Secondary Structure)	242
問題 6.4	系列アライメント (Sequence Alignment)	246
問題 6.5	線形領域の系列アライメント (Sequence Alignment in Linear Space)	253
問題 6.6	最短パス (Shortest Path) (問題 4.4)	258
問題 6.7	グラフの負閉路 (Negative Cycles in a Graph)	269
問題 7.1	最大フロー (Maximum-Flow)	302
問題 7.2	二部グラフのマッチング (Bipartite Matching) (問題 1.4)	330
問題 7.3	辺素パス (Edge-Disjoint Paths)	336
問題 7.4	需要付き循環フロー (Circulation with Demands)	340
問題 7.5	下界のある需要付き循環フロー (Circulation with Demands and Lower Bounds)	344
問題 7.6	市場調査デザイン (Survey Design)	346
問題 7.7	航空スケジューリング (Airline Scheduling)	348
問題 7.8	画像分割 (Image Segmentation)	353

問題 7.9	プロジェクト選択 (Project Selection)	356
問題 7.10	野球優勝可能性消去 (Baseball Elimination)	360
問題 7.11	最小コスト完全マッチング (Minimum-Cost Perfect Matching)	364
問題 8.1	独立集合 (Independent Set) (問題 1.5)	410
問題 8.2	点カバー (Vertex Cover)	411
問題 8.3	集合カバー (Set Cover)	412
問題 8.4	集合パッキング (Set Packing)	414
問題 8.5	充足可能性 (Satisfiability)	415
問題 8.6	3-充足可能性 (3-Satisfiability)	416
問題 8.7	回路充足可能性 (Circuit Satisfiability)	423
問題 8.8	巡回セールスマン (Traveling Salesman)	430
問題 8.9	ハミルトン閉路 (Hamiltonian Cycle)	431
問題 8.10	ハミルトンパス (Hamiltonian Path)	435
問題 8.11	3次元マッチング (3-Dimensional Matching)	437
問題 8.12	グラフ彩色 (Graph Coloring)	441
問題 8.13	部分集合和 (Subset Sum) (問題 6.2)	446
問題 8.14	開始時刻と期限をもつスケジューリング (Scheduling with Release Times and Deadlines)	448
問題 9.1	計画 (Planning)	487, 492
問題 9.2	限量化 3-SAT (Quantified 3-SAT)	488
問題 9.3	ゲーム (Game)	489
問題 10.1	$k$ -点カバー ( $k$ -Vertex Cover)	506
問題 10.2	円弧彩色 (Coloring Circular Arcs)	514
問題 10.3	木分解 (Tree Decomposition)	532
問題 11.1	負荷均等化 (Load Balancing)	548
問題 11.2	センター選択 (Center Selection)	553
問題 11.3	集合カバー (Set Cover) (問題 8.3)	559
問題 11.4	点カバー (Vertex Cover) (問題 8.2)	564
問題 11.5	最大素パス (Maximum Disjoint Paths)	569
問題 11.6	一般化負荷均等化 (Generalized Load Balancing)	581
問題 11.7	ナップザック (Knapsack)	587
問題 12.1	Hopfield ニューラルネットワークの安定状態図 (Stable Configuration in Hopfield Neural Network)	612
問題 12.2	最大カット (Maximum Cut)	616
問題 12.3	ラベル付け (Labeling)	621
問題 12.4	Nash 均衡解 (Nash Equilibrium)	629
問題 13.1	競合の解消 (Contention Resolution)	646
問題 13.2	大域的最小カット (Global Minimum Cut)	651
問題 13.3	MAX 3-SAT	661
問題 13.4	中央値発見 (Finding the Median)	664
問題 13.5	ハッシング (Hashing)	670
問題 13.6	最近点对 (Closest Pair of Points) (問題 5.2)	677
問題 13.7	キャッシュ管理 (Cache Maintenance) (問題 4.3)	684
問題 13.8	負荷均等化 (Load Balancing) (問題 11.1)	694
問題 13.9	パケットルーティング (Packet Routing)	696
問題 14.1	インターネットにおける高速パケットスイッチング (High-Speed Packet Switching on the Internet)	726



# 詳細な解説付きアルゴリズムの一覧

アルゴリズム 1.1	安定マッチングアルゴリズム (Stable Matching Algorithm)	5
アルゴリズム 3.1	二部グラフ性判定アルゴリズム (Testing Bipartiteness)	86
アルゴリズム 3.2	トポロジカルソート (Topological Sort)	92
アルゴリズム 4.1	区間スケジューリングに対するグリーディアルゴリズム (Greedy Interval Scheduling Algorithm)	104
アルゴリズム 4.2	区間分割アルゴリズム (Interval Partitioning Algorithm)	111
アルゴリズム 4.3	最大遅延最小化スケジューリングアルゴリズム (Scheduling Algorithm for Minimizing Maximum Lateness)	114
アルゴリズム 4.4	最適キャッシングアルゴリズム (Optimal Caching Algorithm)	120
アルゴリズム 4.5	Dijkstra のアルゴリズム (Dijkstra's Algorithm)	125
アルゴリズム 4.6	最小全点木アルゴリズム (Minimum Spanning Tree Algorithm)	130
アルゴリズム 4.7	クラスタリングアルゴリズム (Clustering Algorithm)	145
アルゴリズム 4.8	最適な接頭語符号を構築する Huffman のアルゴリズム (Huffman's Optimal Prefix Code Algorithm)	151
アルゴリズム 4.9	最小コスト有向木アルゴリズム (Minimum-Cost Arborescence Algorithm)	162
アルゴリズム 5.1	反転数の数え上げアルゴリズム (Counting Inversions Algorithm)	198
アルゴリズム 5.2	最近点对発見アルゴリズム (Closest Pair Algorithm)	202
アルゴリズム 5.3	整数積アルゴリズム (Integer Multiplication Algorithm)	207
アルゴリズム 5.4	畳込み計算アルゴリズム (Convolution Algorithm)	212
アルゴリズム 6.1	重み付き区間スケジューリングアルゴリズム：再帰版 (Weighted Interval Scheduling Algorithm)	224
アルゴリズム 6.2	重み付き区間スケジューリングアルゴリズム：反復版 (Weighted Interval Scheduling Algorithm: Iteration)	230
アルゴリズム 6.3	区分最小二乗法 (Segmented Least Squares Algorithm)	234
アルゴリズム 6.4	部分集合和アルゴリズム (Subset Sum Algorithm)	238
アルゴリズム 6.5	RNA 二次構造アルゴリズム (RNA Secondary Structure Algorithm)	244
アルゴリズム 6.6	系列アライメントアルゴリズム (Sequence Alignment Algorithm)	249
アルゴリズム 6.7	線形領域の系列アライメントアルゴリズム (Sequence Alignment Algorithm in Linear Space)	253
アルゴリズム 6.8	最短パスアルゴリズム：動的計画法版 (Shortest Path: Dynamic Programming Algorithm)	259
アルゴリズム 6.9	負閉路発見アルゴリズム (Negative Cycle Algorithm)	270
アルゴリズム 7.1	Ford-Fulkerson アルゴリズム (Ford-Fulkerson Algorithm)	305
アルゴリズム 7.2	スケーリング最大フローアルゴリズム (Scaling Max-Flow Algorithm)	317
アルゴリズム 7.3	プリフロー・プッシュ最大フローアルゴリズム (Preflow-Push Maximum-Flow Algorithm)	320
アルゴリズム 7.4	二部グラフのマッチングアルゴリズム	

	(Bipartite Matching Algorithm)	330
アルゴリズム 7.5	辺素パスアルゴリズム (Edge-Disjoint Paths Algorithm)	336
アルゴリズム 7.6	需要付き循環フローアルゴリズム (Algorithm for Circulation with Demands)	342
アルゴリズム 7.7	下界のある需要付き循環フローアルゴリズム (Algorithm for Circulation with Demands and Lower Bounds)	344
アルゴリズム 7.8	市場調査デザインアルゴリズム (Survey Design Algorithm)	347
アルゴリズム 7.9	航空スケジューリングアルゴリズム (Airline Scheduling Algorithm)	350
アルゴリズム 7.10	画像分割アルゴリズム (Image Segmentation Algorithm)	353
アルゴリズム 7.11	プロジェクト選択アルゴリズム (Project Selection Algorithm)	357
アルゴリズム 7.12	野球優勝可能性消去アルゴリズム (Baseball Elimination Algorithm)	361
アルゴリズム 7.13	最小コスト完全マッチングアルゴリズム (Minimum-Cost Perfect Matching Algorithm)	364
アルゴリズム 9.1	QSAT に対するアルゴリズム (Algorithm for QSAT)	490
アルゴリズム 9.2	計画アルゴリズム (Planning Algorithm)	493
アルゴリズム 10.1	$k$ -点カバーアルゴリズム ( $k$ -Vertex Cover Algorithm)	507
アルゴリズム 10.2	円弧彩色アルゴリズム (Circular-Arc Coloring Algorithm)	516
アルゴリズム 10.3	木分解アルゴリズム (Tree Decomposition Algorithm)	532
アルゴリズム 11.1	負荷均等化アルゴリズム (Load Balancing Algorithm)	548
アルゴリズム 11.2	センター選択アルゴリズム (Center Selection Algorithm)	554
アルゴリズム 11.3	集合カバーアルゴリズム (Set Cover Algorithm)	559
アルゴリズム 11.4	点カバーアルゴリズム：価格付け法 (Vertex Cover Algorithm: Pricing Method)	566
アルゴリズム 11.5	最大素パスアルゴリズム (Maximum Disjoint Paths Algorithm)	571
アルゴリズム 11.6	価格付け法アルゴリズム (Pricing Algorithm)	573
アルゴリズム 11.7	一般化負荷均等化アルゴリズム (Generalized Load Balancing Algorithm)	582
アルゴリズム 11.8	ナップサック問題に対する多項式時間近似スキーム (Polynomial-Time Approximation Scheme for Knapsack Problem)	588
アルゴリズム 12.1	Hopfield ニューラルネットワークの安定状態図を求めるアルゴリズム (Algorithm for Stable Configuration in Hopfield Neural Network)	613
アルゴリズム 12.2	最大カットアルゴリズム (Maximum Cut Algorithm)	617
アルゴリズム 12.3	ラベル付けアルゴリズム (Labeling Algorithm)	623
アルゴリズム 13.1	競合解消に対する乱択アルゴリズム (Randomized Algorithm for Contention Resolution)	647
アルゴリズム 13.2	大域的最小カットに対する乱択アルゴリズム (Randomized Algorithm for Global Minimum Cut)	652
アルゴリズム 13.3	MAX 3-SAT に対する乱択近似アルゴリズム (Randomized Approximation Algorithm for MAX 3-SAT)	661
アルゴリズム 13.4	中央値発見アルゴリズム (Algorithm for Finding the Median)	664
アルゴリズム 13.5	ハッシュデータ構造の設計 (Designing Data Structure for Hashing)	671
アルゴリズム 13.6	最近点对問題に対する乱択アプローチ (Randomized Approach for Closest Pair of Points)	678
アルゴリズム 13.7	マーク付けアルゴリズム (Marking Algorithm)	686
アルゴリズム 13.8	乱択マーク付けアルゴリズム (Randomized Marking Algorithm)	689
アルゴリズム 13.9	パケットルーティングに対する乱択スケジューリングアルゴリズム (Randomized Scheduling Algorithm for Packet Routing)	698
アルゴリズム 14.1	純粋出力キューイングアルゴリズム (Pure Output Queuing Algorithm)	729

# 序 文

アルゴリズム的な考え方は世の中に広く浸透してきている。実際、それは、情報科学分野はもちろん、実社会の様々な状況から見ても明らかである。インターネットのルーティングプロトコルの標準化において、大きな変更は何度か行われてきたが、それは、ある最短パスアルゴリズムの欠点と、他のアルゴリズムの相対的な利点とを論議した結果としてとらえることもできる。遺伝子やゲノムの類似性を表現するのに生物学者が用いている基本的な概念も、アルゴリズム的な定義を有している。現実の組合せ的オークションの実行可能性において経済学者が心配する事項は、部分的には、これらのオークションが、特殊な場合に、計算不可能となる探索問題を含むことに起因している。このように、アルゴリズム的な概念は、伝統的な由来の有名な問題にのみ存在するのではなく、広範囲にわたる分野の至るところで生じている新しい問題において、ごく普通に反映されているのである。ユーザーに広告を提供する Yahoo! 社のシステムは、本質的には、ネットワークフローを用いてモデル化して記述されている、と Yahoo! 社の科学者がある日に昼食をとりながら話してくれた。また、ニューヨークへの旅行の途中でたまたま出会った人からも、学生時代にアルゴリズムを学んでいたおかげで、現在は、大病院における要員計画の経営コンサルタントとして働いている、ということを知っている。

アルゴリズムに多くの応用があることのみが重要であるというわけではない。より重要なのは、情報科学の全分野において、アルゴリズム的な考え方により、通常では見えなかったものまでが見えてくるようになる点である。このように、アルゴリズム的な考え方は強力なレンズとしての役割を果たしているということが、より深い意味で重要なのである。アルゴリズム的な問題が情報科学の核心となっていることは確かであるが、それらは、きれいにまとめられて数学的に正確な形をした問題としてやってくることは極めてまれである。むしろ、煩雑で特定の応用に関する細部が大量に付随してくる傾向があり、その中には本質的なものも余分なものもあったりする。結果として、アルゴリズム設計の実際的な作業は、問題の中の、数学的にきれいな核となる部分を見出す仕事と、問題の構造に基づいた適切なアルゴリズム設計技法を見極める仕事という、二つの基本的な構成要素からなっている。これら二つの構成要素は相互に関連し合っている。すなわち、利用できる様々な設計技法に熟練すればするほど、煩雑な情報の氾濫する問題に潜んでいるきれいな定式化を引き出すことができるようになる。このように、アルゴリズム的な考え方の効用は、設定が整った問題の解の獲得のみにとどまるものではない。むしろ、アルゴリズム的な考え方に基づいて、潜んでいる問題を明快に表現する言語が獲得で

き、そしてさらなる展開への扉が開けるといふ点に、アルゴリズム的な考え方の最大の効用があるのである。

したがって、本書では、様々な応用分野における計算が必要な問題に対して、まずその問題を定式化することから始めて、複数あるアルゴリズム設計技法からどの技法を用いるべきかをきちんと判断できる力を確立し、そして最終的にこれらの問題に対する効率的な解法を構築することが、アルゴリズムを設計するプロセス(過程)であるという立場をとって、アルゴリズムにアプローチしていく。情報科学におけるアルゴリズム的な考え方の役割を幅広く探求していき、これらの考え方を、アルゴリズムを設計し解析できるように正確に定式化された問題に関連づけていく。言い換えれば、これらの問題の動機付けとなっている潜在的な問題点とは何なのか、そしてそれらを定式化する特定の方法はどのようにすれば選べるのか、あるいは、様々な状況においてどの設計原理を適用すれば適切なのか、ということを決めていく。

このようなことを考慮して、本書の目標は、様々な分野で生じる複雑な形の計算を伴う問題からアルゴリズム的な問題の明快な定式化を発見する方法と、その定式化に基づいて実際の問題に対する効率的なアルゴリズムを設計する方法を助言し、提供することであると言える。さらに、洗練されたアルゴリズムを理解するには、失敗や袋小路に陥る単純な最初のアプローチとともに、最終的に正解に至るアプローチまでを与えて、用いる一連のアイデアを省略せずに説明するのが最も良いということもしばしば経験されている。したがって、本書のアルゴリズム記述の形式は、必ずしも問題の記述からアルゴリズムに至る最短の道とはなっていないことも多い。著者らは同僚とともにこれらの点について熟考した結果、特定のある問題に対してはこのやり方がより良いと確信している。

## 本書の概観

本書は、(『CS1』と『CS2』に対応する)プログラミング演習付きの標準的な情報科学基礎の2科目を履修し、基礎的なアルゴリズム、木やグラフなどの操作、配列、リスト、キュー、スタックなどの基本的なデータ構造のプログラムを作成して実行した経験を有する学生を念頭に置いている。『CS1/CS2』の両科目から『アルゴリズム基礎』の科目への橋渡しは、完全には標準化されていないので、本書での各トピックの扱いは、自己完結する形で取り上げている。すなわち、本書で取り上げている入門的な部分のトピックに関して、『CS1/CS2』の両科目ですでに学習済みの大学もあるだろうし、『アルゴリズム基礎』の科目に入っている大学もあると思われる。したがって、そのようなトピックは、復習として取り上げることでもできるし、新しいトピックとして取り上げることもできる。本書にそのようなトピックを含めたのは、予備知識を前提にせず、より広範囲の科目に柔軟に対応できるようにと願っているからである。

上記の点を考慮して、情報科学関連の広い分野にまたがる問題を取り上げてアルゴリズム設計技法を展開している。実際、代表的な例を数例挙げると、システムやネットワークからの応用で生じる問題(キャッシング、スイッチング、インターネットでの内部ドメインルーティング)、人工知能から生じる問題(計画問題、ゲーム、Hopfield ネットワーク)、コンピュータビジョンからの問題(画像分割)、データマイニング(変更点検出)、オペレーションズリサーチ(航空スケジューリング)、計算生物学(系列アライメント、RNA 二次構造)などを本書で取り上げている。

計算困難性の概念，特に，NP-完全性の概念は，本書では大きい役割を果たしている．これは，アルゴリズム設計のプロセスに対する著者らの一貫する信念である．実用上生じる問題は，効率的に解を求められるときもあるが，一方で NP-完全であるときもある．したがって，新しく直面した問題に対して，その問題を本当にまじめに取り上げるのであれば，どちらの状況になったとしても対処できるようにしておくことが必要となる．実用から自然に生じる情報科学の問題は NP-完全であることが多いので，そのような計算困難問題にうまく対応できる方法を開発することが，アルゴリズムの研究分野では極めて重要なテーマになっている．本書では，このテーマを真剣に取り上げて議論している．実際，NP-完全であるような問題に対して，NP-完全であることを発見することが最終目標なのではなく，発見後はさらに，近似アルゴリズム，局所探索のヒューリスティックス，あるいは，計算が容易となる特別な構造の可能性，を追求することが大切であると考えている．したがって，本書では，これら三つのアプローチを詳細に取り上げ，議論している．

#### 演習問題と解答付き演習問題

本書の特徴の一つは演習問題にある．すべての章に演習問題があり，全体で 200 以上になっている．これらの演習問題は，ほとんどすべてが Cornell 大学の授業の一環として，実際にレポート課題や試験問題として取り上げたものである．演習問題は，本書の最も重要な構成要素であり，各章で取り上げたテーマに沿うように段階的に構成されている．情報科学やそれ以外の関係する分野の応用から生じた問題が多いので，問題の詳細な背景も説明している．したがって，上述の訓練もできる．すなわち，最初に問題の本質を抽出して必要な記法を用いて数学的に定式化し，次にアルゴリズムを設計し，そして最後に，そのアルゴリズムの解析を行って正当性を証明する訓練ができるようにしている（これらの演習問題のいずれに対しても，完全な解答は，詳細なアルゴリズム，計算時間の解析，正当性の証明からなるものと考えている）．これらの問題は，様々な分野の研究者との長年にわたる議論を通じて選ばれたものである．また，ある問題に対しては，副産物として，（もちろん努力すれば可能ではあるが）これまでどこにもなかった興味深いアルゴリズムを記述して残すこともできるようになったと言える．

演習問題にどのように解答すればよいか迷うことがあるかもしれない．そこで，その手助けとして，各章に“解答付き演習問題”を 1 題あるいは 2 題（第 4 章では 3 題）用意し，解答の書き方について述べている．これらの解答付き演習問題の解答では，演習問題の正解として要求される以上のことを書き記している（言い換えると，演習問題がレポート課題として出されたときに要求される満点の解答より，本質的に極めて詳細なものになっている）．本書の他の節でも同様であるが，“解答付き演習問題”での議論は，この種の問題を深く考えて正解を記述することができるような感性の育成を目的としていると見なしてもらえばよいだろう．

これらの演習問題を授業でレポート課題として用いる際の注意を 2 点述べておきたい．第一に，問題はほぼ難易度順に並べられている．もちろん，その並べ方はおおよその基準に基づくものであるので，その順番を重視しすぎないようにお願いしたい．実際，ほぼすべての問題が自習用のレポート課題として作成したものであり，したがって，各章の大部分の問題は，ほぼ同程度の難易度である．第二に，最初の数題を除いて，演習問題はある程度の時間をかけないと解けないような問題になっていることである．すなわち，そのような問題に対しては，その

章で取り上げたアルゴリズム技法とその問題の定式化との関係をまず明らかにし、その後、必要とされるアルゴリズムを実際に設計することに、ある程度の時間をかけることが要求されている。Cornell 大学の学部の授業では、毎週これらの問題から 3 題程度をレポート課題としていた。

### 教育的工夫と補助教材

本書は、演習問題と解答付き演習問題のほかにも、授業を進める上で役に立つ教育的工夫と補助教材も多数備えている。

前にも注意したように、本書では、極めて多くの節で、アルゴリズム的な問題の定式化、特にその問題の背景と動機付けも含めた定式化、およびその問題に対するアルゴリズムの設計と解析に力点を置いた記述をしている。このスタイルを明快にするために、これらの節は、本書全体のほぼすべてで、以下のように同一のセクション構成をとっている。最初に、“問題”のセクションでは、問題の背景を説明しながら正確な定式化を与えている。次に、“アルゴリズム”のセクションでは、適切なアルゴリズム技法を用いてアルゴリズムを実際に設計している。最後に、“アルゴリズムの解析”のセクションでは、設計したアルゴリズムの性質を明らかにし、アルゴリズムの計算時間を解析している。本書では、これらのセクションを鳥の羽のアイコンでハイライトしている。問題の拡張やアルゴリズムのさらなる解析が望まれるときは、そのためにさらにセクションを加えている。この構造により、様々な応用分野の計算問題に対して、問題の背景と定式化から最終的にその問題の解決方法とその動作の詳細な解析までを、かなり統一的なスタイルで与えるという目標が達成できていると確信している。

本書をサポートする補助教材も多数利用できる。すべての問題に対する解答付きの教員用マニュアルも利用できる。さらに、Princeton 大学の Kevin Wayne が開発した講義スライドも利用できる。これらの講義スライドは本書の節の順に沿って構成されていて、したがって、本書を教科書とする講義の基礎教材としても用いることができる。これらのファイルは [www.aw.com](http://www.aw.com) からダウンロードできる。教師用のログインアカウントやパスワードは “Kleinberg” あるいは “Tardos” のホームページを探すが、近くの Addison-Wesley 社に連絡すれば獲得できる<sup>1</sup>。

最後に、本書に対する感想やコメントを歓迎する。特に、これほど長い本ではよくあることであるが、本書にも誤りが残っているものと思われる。誤りに対するコメント等は、[algbook@cs.cornell.edu](mailto:algbook@cs.cornell.edu) にメールで知らせてもらえればと思っている。その際には、メッセージ行(件名)に “feedback” と書いてほしい。

### 各章の概要

第 1 章は代表的なアルゴリズム的問題の紹介から始めている。そこでは、はじめに安定マッチング問題を取り上げている。この問題を利用することで、他の抽象的な議論を用いるよりも、アルゴリズム設計における基本的な問題点の提起がより具体的になり、明快で華麗にできるようになると確信しているからである。実際、安定マッチング問題は、複雑な実世界から生じた問題であるにもかかわらず、極めて自然な問題であり、したがって、問題の定式化も興味深い

<sup>1</sup> 訳注：原書の日本語版に関しては共立出版に連絡すること。

形で記述でき、それを解く有効なアルゴリズムも驚くほど印象的に提示できる。第1章の残りの部分では、本書の後ろの章で取り上げるトピックと深く関係している五つの“代表的な問題”を議論している。これらの五つの問題は、独立集合問題の変種版あるいは特殊版と見なすことができ、互いに密接に関係している。それらのうち、一つはグリーディアルゴリズムで解け、一つは動的計画法で解け、一つはネットワークフローで解け、一つは独立集合問題そのものでNP-完全であり、最後の一つはPSPACE-完全である。密接に関係する問題でありながら、このように計算の複雑さが大きく異なるということが本書の重要なテーマとなる点であり、したがって、本書での進行に伴い、これらの五つの問題はそれぞれ道標として働くことになる。

第2章と第3章は、前述したように講義科目『CS1/CS2』からの橋渡しの役割を果たしている。第2章では、アルゴリズム解析で用いられるキーとなる数学的な定義と記法を、それらの背景の動機付けになっている原理とともに紹介している。問題が計算容易である（効率的に解ける）ということが何を意味するのかを形式ばらずに概観し、多項式時間の概念を計算容易性（効率性）の正式な概念として採用している。そして、関数の増加率と漸近的解析をより形式的に議論している。さらに、アルゴリズム解析でよく現れる関数を、その関数が実際に現れる標準的な問題とともに紹介している。第3章では、本書の多くの問題で現れるグラフの基礎概念とグラフを扱う原始的アルゴリズム操作を取り上げている。講義科目『CS1/CS2』でも、最後のあたりで基本的なグラフアルゴリズムを複数実装する演習があると考えられるが、本書のように、より広いアルゴリズム設計の枠組みで基本的グラフアルゴリズムを取り上げることも重要であると考えている。特に、基本的なグラフの概念の定義、幅優先探索と深さ優先探索のグラフ走査技法、強連結性とトポロジカル順序を含む有向グラフの概念を議論している。

第2章と第3章では、本書全体でアルゴリズムを実装する際に用いる基本的なデータ構造も多数取り上げている。より高度なデータ構造については、後続の章で取り上げている。本書では、基本的には、本書のアルゴリズムを実装するときに必要となったデータ構造をそのとき説明するというスタイルで、データ構造を取り上げている。したがって、第2章と第3章で取り上げているデータ構造は極めて基本的なもので、多くが講義科目『CS1/CS2』でも説明済みのものとなっているが、それらは後ろの章のアルゴリズムで頻繁に用いられるからである。すなわち、本書では、より広いアルゴリズムの設計と解析の枠組みでこれらのデータ構造に焦点を当てている。

第4章から第7章までは、アルゴリズム設計の主たる技法を取り上げている。すなわち、グリーディアルゴリズム、分割統治法、動的計画法、ネットワークフローを取り上げている。グリーディアルゴリズムでは、それが正しく機能するときとうまく機能しないときとを識別することが挑戦すべき課題となる。第4章では、グリーディアルゴリズムが正しく機能することの証明で用いられる議論を整理して分類することが中心となっている。グリーディアルゴリズムの応用についても第4章の後半で取り上げている。具体的には、最短パス、無向グラフの全点木と有向グラフの全点木、クラスタリング、圧縮への応用を取り上げている。第5章の分割統治法では、計算時間の限界を表す漸化式を解く戦略についてまず議論している。その後、このような漸化式の解法を用いて、様々な問題に対して、素朴な解法を改善するアルゴリズム設計について示している。より具体的には、二つのランク付けの比較、平面上の最近点对の発見、高速フーリエ変換を取り上げている。第6章の動的計画法では、問題に内在して単純に得られる漸化式から始めている。さらに、自然に生じる応用問題を取り上げ、より高度な漸化式を得

る方法を示している。そして第 6 章の最後に、二つの基本的な問題に対して、動的計画法のアプローチを詳細に取り上げている。より具体的には、計算生物学への応用を有する系列アライメントとインターネットのルーティングプロトコルに関するグラフの最短パスに対して、動的計画法のアプローチを議論している。第 7 章では、ネットワークフロー問題に対するアルゴリズムを説明するとともに、フローを応用できる問題を多数取り上げ、議論している。アルゴリズムの講義で取り上げられるネットワークフローでは、通常、その広範にわたる問題への応用については、学生自身に任されてしまうことが多いのであるが、本書ではそれをあえて避けている。すなわち、負荷均等化、スケジューリング、画像分割、およびそれ以外の多数の問題への応用を実際に示して、フローの応用可能性が極めて広範囲にわたることを議論している。

第 8 章と第 9 章では計算困難性を取り上げている。主として NP-完全性について議論しているが、基本的な NP-完全問題をテーマ別に組織化して与えている。すなわち、新しい問題に遭遇してその NP-完全性を証明する際のリダクションの候補を適切に選べるように工夫している。かなり複雑な NP-完全性の証明も取り上げ、困難なリダクションを構成するための方法とそのアイデアを与えている。第 10 章では、さらに、NP-完全性を超える計算困難性の PSPACE-完全性のトピックを取り上げている。これにより、計算困難性は決して NP-完全性で終わるのではないことが強調できる。実際、PSPACE-完全性は、人工知能の分野における計画問題やゲームで中心となる概念の基盤となっている。本書では、PSPACE-完全性を含めて問題の計算困難性に基づく景観を整理し分類しているが、こうしてはじめて人工知能の分野の問題を議論できるようになるのである。

第 10 章から第 12 章では、計算困難性に対処する三つの主たる技法を取り上げている。すなわち、特殊構造に基づくアルゴリズム、近似アルゴリズム、および局所探索によるヒューリスティックスを議論している。第 10 章の特殊構造に基づくアルゴリズムでは、実際に生じる NP-完全問題の入力インスタンスにおいて、必ずしも最悪のケースのインスタンスとならないことも多いことに注目している。したがって、ある種の構造を有して、その構造を利用して効率的に解くアルゴリズムを設計できることもある。そこで、木構造に限定された入力インスタンスでは、NP-完全問題がしばしば効率的に解けることを説明する。さらに、グラフの木分解についても詳細に議論する。この木分解のトピックは、学部の講義より大学院の講義に適したものであるが、実用上の有効性と密接に関連する技法である。さらに、現在のところこのトピックを扱っている書籍が極めて少ないので、本書であえて取り上げている。第 11 章の近似アルゴリズムでは、有効なアルゴリズム設計のプロセスを説明するとともに、その近似性能を得るための最適解に対する詳細な議論を行っている。アルゴリズム設計の技法としては、グリーディアルゴリズム、線形計画法、および両者の概念を組み合わせる“価格付け法”と呼んでいる方法に焦点を絞って説明している。第 12 章では、Metropolis アルゴリズムとシミュレーテッドアニーリングを含む局所探索ヒューリスティックスについて議論している。局所探索のトピックは、そのようにして得られたアルゴリズムに対して、証明可能な近似保証がほとんど知られていないこともあり、通常の学部のアルゴリズムの科目では取り上げられることはない。しかし、実際には広く用いられている方法であるので、それらについての知識をもっておくのはよいことであると考えている。そこで本書では、このトピックを取り上げ、そしてある場合には近似保証が証明できることを示している。

第 13 章では、アルゴリズム設計における乱数の使用を取り上げている。ランダム化アルゴ



リズム（本書では乱択アルゴリズムと呼んでいる）は、大学院レベルの良書がいくつかあるトピックである。本書では、学部の離散数学の講義で取り上げられる確率の知識のみを前提知識として仮定して、乱択アルゴリズムについての簡潔な解説を与えている。これにより、乱択アルゴリズムに必要な確率の知識は極めて少なく済むことが理解できる。

### 本書の利用法

本書は、第一義的には、学部の講義科目である『アルゴリズム基礎』の教科書として執筆したものであるが、大学院の講義の入門用の教科書としても用いることができる。

Cornell 大学で学部用として用いるときには、各節をほぼ 1 回の講義で終わるようにして進めてきている。内容的に 1 回では済まない節（たとえば、さらなる応用が付加的な例として挙げられている節など）では、この追加の部分を講義外での自習用の補助材料として取り上げている。節番号に\*（星）印のついている節は授業ではスキップして省略している。もちろん、これらの星印の節は重要なトピックを含んでいるが、主たるテーマから見て中心的なものでもないこともあるし、また内容がより高度になっているときもある。以上の点から、Cornell 大学の学部の授業では、前半の章では、各章で 1, 2 節程度スキップして省略している（たとえば、4.3 節, 4.7 節, 4.8 節, 5.5 節, 5.6 節, 6.5 節, 7.6 節, 7.11 節は省略することが多い）。

最後にもう少しだけ注意しておこう。後半の章を“より高度である”と見なして学部のアルゴリズムの授業から除外するのではなく、むしろそれらの章の最初の数節に関しては、学部の学生も知っておいたほうがよいと考えている。したがって、Cornell 大学の学部の講義では、後半の章の題材も取り上げている。実際、これらのトピックはすべて学部レベルでも重要視してよいものになっている。

第 2 章と第 3 章に関しては、Cornell 大学では『アルゴリズム基礎』以前の科目で取り上げているので、その復習として扱っている。しかしながら、前述したように、大学ごとのカリキュラムに依存するところが大きいので、各大学でこれらの章の扱いを決定すればよいと思われる。

したがって、Cornell 大学では『アルゴリズム基礎』の科目のシラバスはおおよそ以下のよう構成になっている。すなわち、第 1 章, 第 4~8 章（ただし 4.3 節, 4.7~4.9 節, 5.5~5.6 節, 6.5 節, 6.10 節, 7.4 節, 7.6 節, 7.11 節, 7.13 節は除く）、第 9 章（ごく簡単に）、第 10 章の 10.1 節と 10.2 節、第 11 章の 11.1 節, 11.2 節, 11.6 節, 11.8 節、第 12 章の 12.1~12.3 節、第 13 章の 13.1~13.5 節からなる構成となっている。

大学院のアルゴリズムの科目の入門用としてももちろん用いることができる。そのような科目では、アルゴリズム設計における現在の重要研究テーマを紹介して、学生の将来の様々な分野での研究において役立つようにすることが大切であると考えている。そこで、ここでも有用となる問題の定式化に重点を置いている。各分野の学生の研究において、取り上げる問題の定式化はすぐに必要となると考えられるからである。この種の目的の科目内容として、第 4 章と第 6 章の後半のトピック（4.5~4.9 節と 6.5~6.10 節）、前半の節はかなりスピードを上げて取り扱いながら第 7 章の全部、（大学院入学の学生は学部の授業でこのトピックを『アルゴリズム基礎』の後で学ぶことになっているので）第 8 章の NP-完全性をかなりスピードを上げて取り上げ、その後、残りの時間を第 10~13 章に当てるとというシラバスの構成が考えられる。この大学院の入門科目では、より高度な節に焦点を合わせているが、復習や背景となる基礎知識

を得るための完全な書物があると非常に役に立つこともわかっている。実際、そのような入門科目を受講する学生には、背景の予備知識に大きな差があることが多いからである。

最後に、本書は、大学院生、研究者、情報学科の教師の自習用のテキストとしても用いることができる。自分の仕事の枠組みで、ある特定のアルゴリズム設計技法を用いることができるようになる感性を身につけたいと考えている人には、自習用に最適であると考えている。実際、多数の大学院生や同僚がこのような形で本書を用いてきている。

## 謝辞

本書は、Cornell 大学で著者らが担当してきたアルゴリズムの長年の講義の経験から生まれたものである。何年にもわたる講義を経て、この分野の成長と歩調を合わせて取り上げる内容も広がり、洗練されてきている。この科目で取り上げるべき内容の改善および決定については、Cornell 大学の教授陣からその間にいただいた助言に基づくところが大きい。特に、Juris Hartmanis, Monika Henzinger, John Hopcroft, Dexter Kozen, Ronitt Rubinfeld, Sam Toueg から助言をいただいた。さらに、Cornell 大学のすべての同僚からも、取り上げるべき題材やこの分野の本質に関する問題点について、この上ないほどの討論をいただいた。

この講義の授業では、教育補助者 (TA) に題材の提供も含めて限らない協力をいただいた。学部と大学院の教育補助者である Siddharth Alexander, Rie Ando, Elliot Anshelevich, Lars Backstrom, Steve Baker, Ralph Benzinger, John Bicket, Doug Burdick, Mike Connor, Vladimir Dizhoor, Shaddin Doghmi, Alexander Druyan, Bowei Du, Sasha Evfimievski, Ariful Gani, Vadim Grinshpun, Ara Hayrapetyan, Chris Jeuell, Igor Kats, Omar Khan, Mikhail Kobayakov, Alexei Kopylov, Brian Kulis, Amit Kumar, Yeongwee Lee, Henry Lin, Ashwin Machanavaajhala, Ayan Mandal, Bill McCloskey, Leonid Meyerguz, Evan Moran, Niranjana Nagarajan, Tina Nolte, Travis Ortogero, Martin Pál, Jon Peress, Matt Piotrowski, Joe Polastre, Mike Priscott, Xin Qi, Venu Ramasubramanian, Aditya Rao, David Richardson, Brian Sabino, Rachit Siamwalla, Sebastian Silgado, Alex Slivkins, Chaitanya Swamy, Perry Tam, Nadya Travinin, Sergei Vassilvitskii, Matthew Wachs, Tom Wexler, Shan-Leung Maverick Woo, Justin Yang, Misha Zatsman に深く感謝する。彼らの多くが、本書について価値ある洞察、提案、コメントをしてくれた。授業を受講して本書の草稿に対してコメントや感想をくれたすべての学生にも感謝する。

過去数年にわたり、本書の草稿の段階から授業に利用してくれた他大学の先生からの感想や助言も本書の改善に大いに役立った。Anna Karlin は、Washington 大学で彼女の担当する講義の教科書として、本書の草稿を初めて使用してくれた。彼女に続いて、講義の教科書や参考書として使用してくれた Paul Beame, Allan Borodin, Devdatt Dubhashi, David Kempe, Gene Kleinberg, Dexter Kozen, Amit Kumar, Mike Molloy, Yuval Rabani, Tim Roughgarden, Alexa Sharp, Shanghua Teng, Aravind Srinivasan, Dieter van Melkebeek, Kevin Wayne, Tom Wexler, Sue Whitesides からの感想と助言にも感謝したい。彼らの助言に基づいて本書の多くの内容を改訂している。Kevin Wayne は、本書に付随する補助教材まで作成してくれた。彼には特別に感謝したい。これらの補助教材は、本書を使用して講義をする先生の手助けになっていると確信している。

そのほかにも、本書のいくつかのトピックに関しては、その分野の先生や研究者からの助言に基づいたアプローチになっている。助言をくれた Yuri Boykov, Ron Elber, Dan Huttenlocher, Bobby Kleinberg, Evie Kleinberg, Lillian Lee, David McAllester, Mark Newman, Prabhakar Raghavan, Bart Selman, David Shmoys, Steve Strogatz, Olga Veksler, Duncan Watts, Ramin Zabih には特に感謝したい。名前を挙げるができなかった多くの人には、ここでお詫びし、感謝する。

Addison Wesley 社のスタッフと一緒に仕事をできたことは楽しい思い出である。第一に、本書の編集から完成への過程における助言と指導に対して、Matt Goldstein に感謝したい。彼には、本書に対する膨大な査読結果の整理と、それを生かした改善策の具体化において限りない協力をいただいた。本書の初期段階での Susan Hartman との会話も極めて有用であった。Matt と Susan, そして、Addison Wesley 社の Michelle Brown, Marilyn Lloyd, Patty Mahtani, Maite Suarez-Rivas および Windfall Software 社の Paul Anagnostopoulos, Jacqui Scarlott がしてくれた本書の編集、印刷、管理に感謝する。専門家としての本書の構成に尽力してくれた Paul と Jacqui には、さらなる感謝をしたい。Dartmouth Publishing 社のスタッフにもお世話になった。特に、表紙のデザインに対して Joyce Wells に、図の作成に対して Nancy Murphy に、索引に対して Ted Laux に、原稿編集と校正に対して Carol Leyba と Jennifer McClain に感謝する。

貴重な時間を割いて、本書の草稿に詳しく思いやりのある意見をくれた Anselm Blumer (Tufts University), Richard Chang (University of Maryland, Baltimore County), Kevin Compton (University of Michigan), Diane Cook (University of Texas, Arlington), Sariel Har-Peled (University of Illinois, Urbana-Champaign), Sanjeev Khanna (University of Pennsylvania), Philip Klein (Brown University), David Matthias (Ohio State University), Adam Meyerson (UCLA), Michael Mitzenmacher (Harvard University), Stephan Olariu (Old Dominion University), Mohan Paturi (UC San Diego), Edgar Ramos (University of Illinois, Urbana-Champaign), Sanjay Ranka (University of Florida, Gainesville), Leon Reznik (Rochester Institute of Technology), Subhash Suri (UC Santa Barbara), Dieter van Melkebeek (University of Wisconsin, Madison), Bulent Yener (Rensselaer Polytechnic Institute) に感謝する。彼らからのコメントにより、最終版では非常に多くの箇所が改善ができた。

最後に家族全員 (Kleinberg の家族の) Lillian と Alice (Tardos の家族の) David, Rebecca と Amy にも感謝したい。彼らのサポート、忍耐、ここでは記すことのできないもろもろの貢献に心から感謝する。

本書は、1990 年代後半の豊かさの絶頂期に、最初の構想が練られた。その時代になるまでは、情報技術を堪能できるのは伝統的に有名人や天空の世界の住人に限定されていると多くの人に思われたものである (これは単に著者らの想像でしかないかもしれない)。何年かたって、株価も安定し幻想から現実の世界に戻った今日では、その時代を経験したことで情報科学の分野も大きく変化したとも考えられるし、あるいは別の意味でそのままであったとも考えられる。情報技術の分野に対する期待と興奮は、この分野が生まれたときからの特徴であり原動力であるが、現在も強烈で魅惑的である。世の中の情報技術に対する関心も依然として極めて高く、計算技術の応用分野はさらなる新分野へと展開されてきている。したがって、このアルゴリズム

研究の分野に興味をもったすべての学生に対して、各自の目的とするものがどのような計算の応用であったとしても、本書が楽しく学べて有用な道標の役割を果たせるようにと願っている。

Ithaca, 2005

Jon Kleinberg ( ジョン・クラインバーグ )

Éva Tardos ( エーバ・タルドシュ )