Algorithm Design
Jon Kleinberg and Eva Tardos

Table of Contents

1 Introduction: Some Representative Problems
  1.1 A First Problem: Stable Matching
  1.2 Five Representative Problems
  1.3 Solved Exercises
  1.4 Exercises
  1.5 Notes and Further Reading

2 Basics of Algorithms Analysis
  2.1 Computational Tractability
  2.2 Asymptotic Order of Growth Notation
  2.3 Implementing the Stable Matching Algorithm using Lists and Arrays
  2.4 A Survey of Common Running Times
  2.5 A More Complex Data Structure: Priority Queues
  2.6 Solved Exercises
  2.5 Exercises
  2.7 Notes and Further Reading

3 Graphs
  3.1 Basic Definitions and Applications
  3.2 Graph Connectivity and Graph Traversal
  3.3 Implementing Graph Traversal using Queues and Stacks
  3.4 Testing Bipartiteness: An Application of Breadth-First Search
  3.5 Connectivity in Directed Graphs
  3.6 Directed Acyclic Graphs and Topological Ordering
  3.7 Solved Exercises
  3.8 Exercises
  3.9 Notes and Further Reading

4 Greedy Algorithms
  4.1 Interval Scheduling: The Greedy Algorithm Stays Ahead
  4.2 Scheduling to Minimize Lateness: An Exchange Argument
  4.3 Optimal Caching: A More Complex Exchange Argument
  4.4 Shortest Paths in a Graph
  4.5 The Minimum Spanning Tree Problem
  4.6 Implementing Kruskal's Algorithm: The Union-Find Data Structure
  4.7 Clustering
  4.8 Huffman Codes and the Problem of Data Compression
  4.9 (*) Minimum-Cost Arborescences: A Multi-Phase Greedy Algorithm
  4.10 Solved Exercises
  4.11 Exercises
  4.12 Notes and Further Reading

5 Divide and Conquer
5.1 A First Recurrence: The Mergesort Algorithm
5.2 Further Recurrence Relations
5.3 Counting Inversions
5.4 Finding the Closest Pair of Points
5.5 Integer Multiplication
5.6 Convolutions and The Fast Fourier Transform
5.7 Solved Exercises
5.8 Exercises
5.9 Notes and Further Reading
6 Dynamic Programming
6.1 Weighted Interval Scheduling: A Recursive Procedure
6.2 Weighted Interval Scheduling: Iterating over Sub-Problems
6.3 Segmented Least Squares: Multi-way Choices
6.4 Subset Sums and Knapsacks: Adding a Variable
6.5 RNA Secondary Structure: Dynamic Programming Over Intervals
6.6 Sequence Alignment
6.7 Sequence Alignment in Linear Space
6.8 Shortest Paths in a Graph
6.9 Shortest Paths and Distance Vector Protocols
6.10 (*) Negative Cycles in a Graph
6.11 Solved Exercises
6.12 Exercises
6.13 Notes and Further Reading
7 Network Flow
7.1 The Maximum Flow Problem and the Ford-Fulkerson Algorithm
7.2 Maximum Flows and Minimum Cuts in a Network
7.3 Choosing Good Augmenting Paths
7.4 (*) The Preflow-Push Maximum Flow Algorithm
7.5 A First Application: The Bipartite Matching Problem
7.6 Disjoint Paths in Directed and Undirected Graphs
7.7 Extensions to the Maximum Flow Problem
7.8 Survey Design
7.9 Airline Scheduling
7.10 Image Segmentation
7.11 Project Selection
7.12 Baseball Elimination
7.13 (*) A Further Direction: Adding Costs to the Matching Problem
7.14 Solved Exercises
7.15 Exercises
7.16 Notes and Further Reading
8 NP and Computational Intractability
8.1 Polynomial-time Reductions
8.2 Efficient Certification and the Definition of NP
8.3 NP-Complete Problems
8.4 Sequencing Problems
8.5 Partitioning Problems
Table of Contents

8.6 Graph Coloring
8.7 Numerical Problems
8.8 co-NP and the Asymmetry of NP
8.9 A Partial Taxonomy of Hard Problems
8.10 Solved Exercises
8.11 Exercises
8.12 Notes and Further Reading

9 PSPACE: A Class of Problems Beyond NP
9.1 PSPACE
9.2 Some Hard Problems in PSPACE
9.3 Solving Quantified Problems and Games in Polynomial Space
9.4 Solving the Planning Problem in Polynomial Space
9.5 Proving Problems PSPACE-Complete
9.6 Solved Exercises
9.7 Exercises
9.8 For Further Reading

10 Extending the Limits of Tractability
10.1 Finding Small Vertex Covers
10.2 Solving NP-hard Problem on Trees
10.3 Coloring a Set of Circular Arcs
10.4 (*) Tree Decompositions of Graphs
10.5 (*)(*) Constructing a Tree Decomposition
10.6 Solved Exercises
10.7 Exercises
10.8 Notes and Further Reading

11 Approximation Algorithms
11.1 Greedy Algorithms and Bounds on the Optimum: A Load Balancing Problem
11.2 The Center Selection Problem
11.3 Set Cover: A General Greedy Heuristic
11.4 The Pricing Method: Vertex Cover
11.5 Maximization via the Pricing method: The Disjoint Paths Problem
11.6 Linear Programming and Rounding: An Application to Vertex Cover
11.7 (*)(*) Load Balancing Revisited: A More Advanced LP Application
11.8 Arbitrarily Good Approximations: the Knapsack Problem
11.9 Solved Exercises
11.10 Exercises
11.11 Notes and Further Reading

12 Local Search
12.1 The Landscape of an Optimization Problem
12.2 The Metropolis Algorithm and Simulated Annealing
12.3 An Application of Local Search to Hopfield Neural Networks
12.4 Maximum Cut Approximation via Local Search
12.5 Choosing a Neighbor Relation
12.6 (*) Classification via Local Search
12.7 Best-Response Dynamics and Nash Equilibria
12.8 Solved Exercises
12.9 Exercises
12.10 Notes and Further Reading

13 Randomized Algorithms
13.1 A First Application: Contention Resolution
13.2 Finding the Global Minimum Cut
13.3 Random Variables and their Expectations
13.4 A Randomized Approximation Algorithm for MAX-3-SAT
13.5 Randomized Divide-and-Conquer: Median-Finding and Quicksort
13.6 Hashing: A Randomized Implementation of Dictionaries
13.7 Finding the Closest Pair of Points: A Randomized Approach
13.8 Randomized Caching
13.9 Chernoff Bounds
13.10 Load Balancing
13.11 (*) Packet Routing
13.12 Background: Some Basic Probability Definitions
13.13 Solved Exercises
13.14 Exercises
13.15 Notes and Further Reading

Epilogue: Algorithms that Run Forever