1. Record Nr. UNINA9910153150703321

Autore	Levitin Anany
Titolo	Introduction to the design & analysis of algorithms / / Anany Levitin ; international edition contributions by Soumen Mukherjee, Aruf Kumar Bhattacharjee
Pubbl/distr/stampa	Boston : , : Pearson, , 2012
ISBN	9781292014111 9780273764113
Edizione	[Third edition.]
Descrizione fisica	1 online resource (589 pages) : illustrations
Disciplina	005.1
Soggetti	Computer algorithms
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Cover -- Title Page -- Contents -- New to the Third Edition -- Preface -- 1 Introduction -- 1.1 What Is an Algorithm? -- Exercises 1.1 -- 1.2 Fundamentals of Algorithmic Problem Solving -- Understanding the Problem -- Ascertaining the Capabilities of the Computational Device -- Choosing between Exact and Approximate Problem Solving -- Algorithm Design Techniques -- Designing an Algorithm and Data Structures -- Methods of Specifying an Algorithm -- Proving an Algorithm's Correctness -- Analyzing an Algorithm -- Coding an Algorithm -- Exercises 1.2 -- 1.3 Important Problem Types -- Sorting -- Searching -- String Processing -- Graph Problems -- Combinatorial Problems -- Geometric Problems -- Numerical Problems -- Exercises 1.3 -- 1.4 Fundamental Data Structures -- Linear Data Structures -- Graphs -- Trees -- Sets and Dictionaries -- Exercises 1.4 -- Summary -- 2 Fundamentals of the Analysis of Algorithm Efficiency -- 2.1 The Analysis Framework 68 -- Measuring an Input's Size -- Units for Measuring Running Time -- Orders of Growth -- Worst-Case, Best-Case, and Average-Case Efficiencies -- Recapitulation of the Analysis Framework -- Exercises 2.1 -- 2.2 Asymptotic Notations and Basic Efficiency Classes -- Informal Introduction -- O-notation -- -notation -- -notation -- Useful Property Involving the Asymptotic Notations -- Using Limits for Comparing Orders of Growth -- Basic Efficiency Classes -- Exercises 2.2 -- 2.3 Mathematical Analysis of

Nonrecursive Algorithms -- Exercises 2.3 -- 2.4 Mathematical Analysis of Recursive Algorithms -- Exercises 2.4 -- 2.5 Example: Computing the nth Fibonacci Number -- Exercises 2.5 -- 2.6 Empirical Analysis of Algorithms -- Exercises 2.6 -- 2.7 Algorithm Visualization -- Summary -- 3 Brute Force and Exhaustive Search -- 3.1 Selection Sort and Bubble Sort -- Selection Sort -- Bubble Sort -- Exercises 3.1.
3.2 Sequential Search and Brute-Force String Matching -- Sequential Search -- Brute-Force String Matching -- Exercises 3.2 -- 3.3 Closest-Pair and Convex-Hull Problems by Brute Force -- Closest-Pair Problem -- Convex-Hull Problem -- Exercises 3.3 -- 3.4 Exhaustive Search -- Traveling Salesman Problem -- Knapsack Problem -- Assignment Problem -- Exercises 3.4 -- 3.5 Depth-First Search and Breadth-First Search -- Depth-First Search -- Breadth-First Search -- Exercises 3.5 -- Summary -- 4 Decrease-and-Conquer -- 4.1 Insertion Sort -- Exercises 4.1 -- 4.2 Topological Sorting -- Exercises 4.2 -- 4.3 Algorithms for Generating Combinatorial Objects -- Generating Permutations -- Generating Subsets -- Exercises 4.3 -- 4.4 Decrease-by-a-Constant-Factor Algorithms -- Binary Search -- Fake-Coin Problem -- Russian Peasant Multiplication -- Josephus Problem -- Exercises 4.4 -- 4.5 Variable-Size-Decrease Algorithms -- Computing a Median and the -- Interpolation Search -- Searching and Insertion in a Binary Search Tree -- The Game of Nim -- Exercises 4.5 -- Summary -- 5 Divide-and-Conquer -- 5.1 Mergesort -- Exercises 5.1 -- 5.2 Quicksort -- Exercises 5.2 -- 5.3 Binary Tree Traversals and Related Properties -- Exercises 5.3 -- 5.4 Multiplication of Large Integers and Strassen's Matrix Multiplication -- Multiplication of Large Integers -- Strassen's Matrix Multiplication -- Exercises 5.4 -- 5.5 The Closest-Pair and Convex-Hull Problems by Divide-and-Conquer -- The Closest-Pair Problem -- Convex-Hull Problem -- Exercises 5.5 -- Summary -- 6 Transform-and-Conquer -- 6.1 Presorting -- Exercises 6.1 -- 6.2 Gaussian Elimination -- LU Decomposition -- Computing a Matrix Inverse -- Computing a Determinant -- Exercises 6.2 -- 6.3 Balanced Search Trees -- AVL Trees -- 2-3 Trees -- Exercises 6.3 -- 6.4 Heaps and Heapsort -- Notion of the Heap -- Heapsort -- Exercises 6.4.
6.5 Horner's Rule and Binary Exponentiation -- Horner's Rule -- Binary Exponentiation -- Exercises 6.5 -- 6.6 Problem Reduction -- Computing the Least Common Multiple -- Counting Paths in a Graph -- Reduction of Optimization Problems -- Linear Programming -- Reduction to Graph Problems -- Exercises 6.6 -- Summary -- 7 Space and Time Trade-Offs -- 7.1 Sorting by Counting -- Exercises 7.1 -- 7.2 Input Enhancement in String Matching -- Horspool's Algorithm -- Boyer-Moore Algorithm -- Exercises 7.2 -- 7.3 Hashing -- Open Hashing (Separate Chaining) -- Closed Hashing (Open Addressing) -- Exercises 7.3 -- 7.4 B-Trees -- Exercises 7.4 -- Summary -- 8 Dynamic Programming -- 8.1 Three Basic Examples -- Exercises 8.1 -- 8.2 The Knapsack Problem and Memory Functions -- Memory Functions -- Exercises 8.2 -- 8.3 Optimal Binary Search Trees -- Exercises 8.3 -- 8.4 Warshall's and Floyd's Algorithms -- Warshall's Algorithm -- Floyd's Algorithm for the All-Pairs Shortest-Paths Problem -- Exercises 8.4 -- Summary -- 9 Greedy Technique -- 9.1 Prim's Algorithm -- Exercises 9.1 -- 9.2 Kruskal's Algorithm -- Disjoint Subsets and Union-Find Algorithms -- Exercises 9.2 -- 9.3 Dijkstra's Algorithm -- Exercises 9.3 -- 9.4 Huffman Trees and Codes -- Exercises 9.4 -- Summary -- 10 Iterative Improvement -- 10.1 The Simplex Method -- Geometric Interpretation of Linear Programming -- An Outline of the Simplex Method -- Further Notes on the Simplex Method -- Exercises 10.1 -- 10.2 The Maximum-Flow Problem -- Exercises 10.2 -- 10.3

Maximum Matching in Bipartite Graphs -- Exercises 10.3 -- 10.4 The Stable Marriage Problem -- Exercises 10.4 -- Summary -- 11 Limitations of Algorithm Power -- 11.1 Lower-Bound Arguments -- Trivial Lower Bounds -- Information-Theoretic Arguments -- Adversary Arguments -- Problem Reduction -- Exercises 11.1 -- 11.2 Decision Trees.
Decision Trees for Sorting -- Decision Trees for Searching a Sorted Array -- Exercises 11.2 -- 11.3 P, NP, and NP-Complete Problems -- P and NP Problems -- NP-Complete Problems -- Exercises 11.3 -- 11.4 Challenges of Numerical Algorithms -- Exercises 11.4 -- Summary -- 12 Coping with the Limitations of Algorithm Power -- 12.1 Backtracking -- n-Queens Problem -- Hamiltonian Circuit Problem -- Subset-Sum Problem -- General Remarks -- Exercises 12.1 -- 12.2 Branch-and-Bound -- Assignment Problem -- Knapsack Problem -- Traveling Salesman Problem -- Exercises 12.2 -- 12.3 Approximation Algorithms for NP-Hard Problems -- Approximation Algorithms for the Traveling Salesman Problem -- Approximation Algorithms for the Knapsack Problem -- Exercises 12.3 -- 12.4 Algorithms for Solving Nonlinear Equations -- Bisection Method -- Method of False Position -- Newton's Method -- Exercises 12.4 -- Summary -- Epilogue -- APPENDIX A -- Useful Formulas for the Analysis of Algorithms -- Properties of Logarithms -- Combinatorics -- Important Summation Formulas -- Sum Manipulation Rules -- Approximation of a Sum by a Definite Integral -- Floor and Ceiling Formulas -- Miscellaneous -- APPENDIX B -- Short Tutorial on Recurrence Relations -- Sequences and Recurrence Relations -- Methods for Solving Recurrence Relations -- Common Recurrence Types in Algorithm Analysis -- References -- Hints to Exercises -- Index -- Numbers and Symbols.

Sommario/riassunto

Based on a new classification of algorithm design techniques and a clear delineation of analysis methods, Introduction to the Design and Analysis of Algorithms presents the subject in a coherent and innovative manner. Written in a student-friendly style, the book emphasizes the understanding of ideas over excessively formal treatment while thoroughly covering the material required in an introductory algorithms course. Popular puzzles are used to motivate students' interest and strengthen their skills in algorithmic problem solving. Other learning-enhancement features include chapter summaries, hints to the exercises, and a detailed solution manual.