1. Record Nr. UNINA9910454516203321

Autore	Edmonds Jeff <1963->
Titolo	How to think about algorithms / / Jeff Edmonds [[electronic resource]]
Pubbl/distr/stampa	Cambridge : , : Cambridge University Press, , 2008
ISBN	1-107-17584-4 0-511-64579-1 9786612390289 1-282-39028-7 1-139-63726-6 0-511-80824-0 0-511-64988-6 0-511-41278-9 0-511-56800-2 0-511-41370-X
Descrizione fisica	1 online resource (xiii, 448 pages) : digital, PDF file(s)
Disciplina	518/.1
Soggetti	Algorithms - Study and teaching Loops (Group theory) - Study and teaching Invariants - Study and teaching Recursion theory - Study and teaching
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Title from publisher's bibliographic system (viewed on 05 Oct 2015).
Nota di contenuto	Iterative algorithms: measures of progress and loop invariants -- Examples using more-of-the-input loop invariants -- Abstract data types -- Narrowing the search space: binary search -- Iterative sorting algorithms -- Euclid's GCD algorithm -- The loop invariant for lower bounds -- Abstractions, techniques, and theory -- Some simple examples of recursive algorithms -- Recursion on trees -- Recursive images -- Parsing with context-free grammars -- Definition of optimization problems -- Graph search algorithms -- Network flows and linear programming -- Greedy algorithms -- Recursive backtracking -- Dynamic programming algorithms -- Examples of dynamic programs -- Reductions and NP-completeness -- Randomized

	algorithms -- Existential and universal quantifiers -- Time complexity -- Logarithms and exponentials -- Asymptotic growth -- Adding-made-easy approximations -- Recurrence relations -- A formal proof of correctness.
Sommario/riassunto	This textbook, for second- or third-year students of computer science, presents insights, notations, and analogies to help them describe and think about algorithms like an expert, without grinding through lots of formal proof. Solutions to many problems are provided to let students check their progress, while class-tested PowerPoint slides are on the web for anyone running the course. By looking at both the big picture and easy step-by-step methods for developing algorithms, the author guides students around the common pitfalls. He stresses paradigms such as loop invariants and recursion to unify a huge range of algorithms into a few meta-algorithms. The book fosters a deeper understanding of how and why each algorithm works. These insights are presented in a careful and clear way, helping students to think abstractly and preparing them for creating their own innovative ways to solve problems.