# Computational Complexity: A Conceptual Perspective

Oded Goldreich

*Computational Complexity: A Conceptual Perspective* is written by Oded Goldreich, and published by Cambridge University Press,  ISBN 978-0-521-88473-0, 606 pages, $70.

This book pursues topics of computational feasibility with an analytical rigor that puts this work outside the interests of typical undergraduate computer science students or the day-to-day work of professionals working on most software intensive IT systems. However, for the advanced student or inquiring professional working on computational complexity, it is an excellent read.

Chapters cover topics in areas such as P and NP, space complexity, randomness, computational problems that are (or appear) infeasible to solve, pseudo-random generators, and probabilistic proof systems. The introduction nicely summarizes the material covered in the rest of the book and includes a diagram of dependencies between chapter topics. Initial chapters cover preliminary topics as preparation for the rest of the book. These are more than topical or historical summaries but generally not sufficient to fully prepare the reader for later material. Readers should approach this text already competent at undergraduate-level algorithms in areas such as basic analysis, algorithm strategies, fundamental algorithm techniques, and the basics for determining computability. Elective work in P versus NP or advanced analysis would be valuable but that isn't really required.

This rest of the book makes a very good read from front-to-back but it can be read topic-by-topic if desired. None of the chapters strictly rely upon evolving examples covered in earlier chapters.

Chapter content is organized in a detailed outline fashion using a mix of short narratives, equations, definitions, proofs, solutions and guidelines. The material moves smoothly between discussions of the topic, exposition of techniques and posing questions to be considered. The early chapters in particular include short teaching notes. These notes provide useful teaching hints, clarify vocabulary, summarize essential concepts, and provide explanations for questions students may have.

Chapters end with notes and exercises. Notes provide details readers may find interesting but aren't essential to covering key topics. Many notes provide historical information. Exercises cover the chapter content well, vary in the type of question to answered and range in difficulty. The exercises aren't organized by difficulty, solutions are not provided and the publisher does not offer a teaching handbook for this text.

Topics are covered well in the chapter narratives and in the extensive use of equations. To cover the material in depth, readers will need to be comfortable with notations for sets, series and logic. Though these skills are required to thoroughly cover the material, the narratives are well written and complete to a certain level of detail. If one is unfamiliar (or out of practice) with the notations and equations used, the narratives remain informative and sufficiently interesting to satisfy a general level of study. The material can also be effectively covered without working through the theorems, propositions and proofs.

No programming code or pseudo-code is included. As a result, there isn't an obvious path from the concepts presented to candidate implementations in hardware or software. This approach isn't necessarily a short-coming and helps many readers focus on the key ideas and concepts rather than getting bogged down into implementation. However, if you intend to apply this material to building a computing system of some sort, be sure to obtain supporting references to bridge the gap getting to systems development.

As described in the forward, this book is intended as a textbook for advanced undergraduate or graduate courses. It would be well suited for courses in computer science, applied mathematics or similar areas of study. This book is also intended, and would be well used, for a program of self-study. As professional development resource, it will be of less interest to the theorist and more to the applied engineering or applied science professional.

What material the reader will find particularly interesting depends greatly on personal interests or preferences about other books on the topics covered. For example, chapters covering the difficulties of finding solutions versus checking solutions (P versus NP) are well-organized, thorough and interesting. However, for a typical college-level study of this topic, there probably isn't any advantage to the material here over the more commonly used texts by Knuth, Sedgewick, or Cormen et al. Most of Goldreich's other published work is in cryptography. That expertise is brought usefully to bear and produces very interesting, more distinctive, reading in the chapters related to apparently infeasible computing challenges. The chapter titled "The Bright Side of Hardness" skillfully avoids delving into depths only of interest to cryptographers but succeeds at exploring computational infeasibility in ways likely fascinating to readers from a broad range of disciplines.

This book provides very well developed material that should interest advanced students either studying or doing new work on computational complexity. It would also be a valuable text for professionals challenged with solving 'hard' computing problems or intending to exploit these types of problems when designing of new types computing systems.

Reviewed by Brian A. Lawler, Association of Computing Machinery and Johns Hopkins University, brian.lawler@acm.org and blawler@jhu.edu.