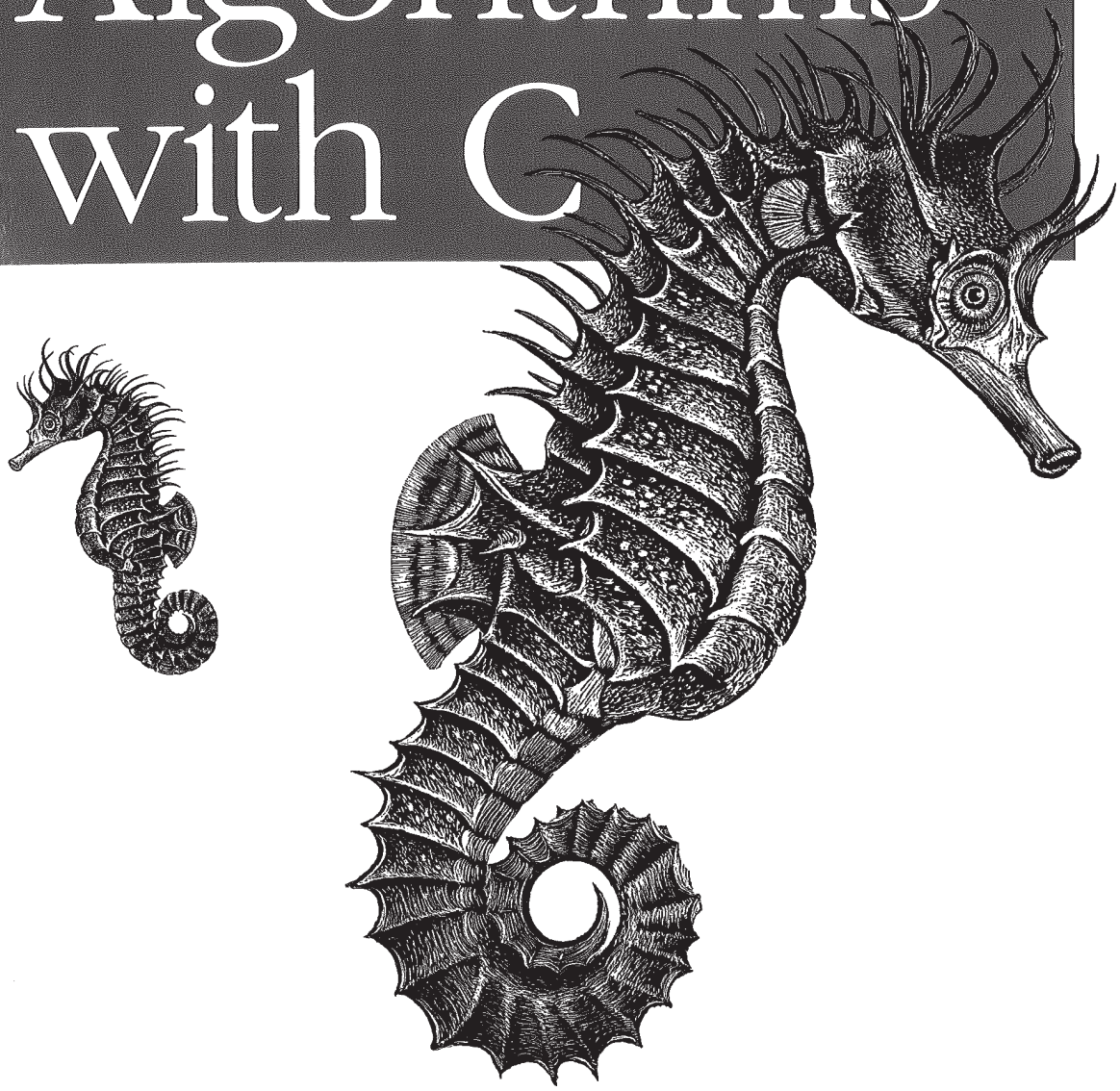
Useful Techniques from Sorting to Encryption

Mastering

Algorithms with C



O'REILLY®

Kyle Loudon

Mastering Algorithms with C

Kyle Loudon

O'REILLY®
Beijing · Cambridge · Farnham · Köln · Sebastopol · Tokyo

Mastering Algorithms with C
by Kyle Loudon

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Editor: Andy Oram

Production Editor: Jeffrey Liggett

Printing History:

August 1999: First Edition.

ISBN: 978-1-565-92453-6
[LSI] [2010-11-30]

14

Data Compression

Data compression is the process of reducing the number of bits used to represent data. It is one of the most significant results of *information theory*, an area of mathematics that addresses various ways to manage and manipulate information. Data compression entails two processes: in one process the data is compressed, or *encoded*, to reduce its size; in a second process it is uncompressed, or *decoded*, to return it to its original state.

To understand why data compression is possible, we must first understand that all data can be characterized by some informational content, called its *entropy* (a term borrowed from thermodynamics). Compression is possible because most data is represented with more bits than its entropy suggests is optimal. To gauge the effectiveness of compression, we look at the ratio of the size of the compressed data divided by its original size, and subtract this from 1. This value is known as the data's *compression ratio*.

In the broadest sense, data compression methods are divided into two classes: *lossy* and *lossless*. In lossy compression we accept a certain loss of accuracy in exchange for greater compression ratios. This is acceptable in some applications, such as graphics and sound processing, provided the degradation is managed carefully. However, frequently we use lossless compression, which ensures that an exact copy of the original data is reproduced when uncompressed.

This chapter focuses on lossless compression, for which there are two general approaches: *minimum redundancy coding* and *dictionary-based methods*. Minimum redundancy coding achieves compression by encoding symbols that occur with great frequency using fewer bits than for those that occur less often. Dictionary-based methods encode data in terms of tokens that take the place of redundant phrases. Example 14-1 is a header for the compression methods presented in this chapter.

This chapter covers:

Bit operations

An important part of data compression because most methods require operating on data one bit at a time to some degree. C provides a number of bitwise operators that can be used to implement an extended class of bit operations.

Huffman coding

One of the oldest and most elegant forms of compression based on minimum redundancy coding. Fundamental to Huffman coding is the construction of a Huffman tree, which is used both to encode and decode the data. Huffman coding is not the most effective form of compression, but it runs fast both when compressing and uncompressing data.

LZ77 (Lempel-Ziv-1977)

One of the fundamental methods of dictionary-based compression. LZ77 uses a sliding window and a look-ahead buffer to encode symbols in terms of phrases encountered earlier in the data. LZ77 generally results in better compression ratios than Huffman coding, but with longer compression times. However, uncompressing data is generally very fast.

Some applications of lossless data compression are:

Software distribution

The process of delivering software on various media. When distributing software on physical media, such as compact discs or magnetic tapes and diskettes, reducing the amount of storage required can produce considerable cost savings in mass distributions.

Archiving

Collecting groups of files into organized libraries. Typically, archives contain large amounts of data. Thus, after creating archives, frequently we compress them.

Mobile computing

An area of computing in which devices typically have limited amounts of memory and secondary storage. Mobile computing generally refers to computing with small, portable devices such as advanced programmable calculators, electronic organizers, and other personal computing devices.

Optimized networking (illustrated in this chapter)

Compression is used especially when sending large amounts of data across wide-area networks. Bandwidth at certain points along wide-area networks is often limited. Although compressing and uncompressing data does require time, in many network applications the cost is well justified.

Embedded applications

An area of computing similar to mobile computing in that devices typically have somewhat limited amounts of memory and secondary storage. Examples of

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.