CAMBRIDGE

Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

Compact Data Structures

A Practical Approach

Compact data structures help represent data in reduced space while allowing querying, navigating, and operating it in compressed form. They are essential tools for efficiently handling massive amounts of data by exploiting the memory hierarchy. They also reduce the resources needed in distributed deployments and make better use of the limited memory in low-end devices.

The field has developed rapidly, reaching a level of maturity that allows practitioners and researchers in application areas to benefit from the use of compact data structures. This first comprehensive book on the topic focuses on the structures that are most relevant for practical use. Readers will learn how the structures work, how to choose the right ones for their application scenario, and how to implement them. Researchers and students in the area will find in the book a definitive guide to the state of the art in compact data structures.

Gonzalo Navarro is Professor of Computer Science at the University of Chile. He has worked for 20 years on the relation between compression and data structures. He has directed or participated in numerous large projects on web research, information retrieval, compressed data structures, and bioinformatics. He is the Editor in Chief of the *ACM Journal of Experimental Algorithmics* and also a member of the editorial board of the journals *Information Retrieval* and *Information Systems*. His publications include the book *Flexible Pattern Matching in Strings* (with M. Raffinot), 20 book chapters, more than 100 journal papers and 200 conference papers; he has also chaired eight international conferences.

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE
Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

Compact Data Structures

A Practical Approach

Gonzalo Navarro

Department of Computer Science,
University of Chile

CAMBRIDGE
UNIVERSITY PRESS

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE
UNIVERSITY PRESS

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781107152380

© Gonzalo Navarro 2016

This publication is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

First published 2016

Printed in the United States of America by Sheridan Books, Inc.

A catalogue record for this publication is available from the British Library.

Library of Congress Cataloging-in-Publication Data
Names: Navarro, Gonzalo, 1969– author.
Title: Compact data structures : a practical approach / Gonzalo Navarro, Universidad de Chile.
Description: New York, NY : University of Cambridge, [2016] | Includes bibliographical references and index.
Identifiers: LCCN 2016023641 | ISBN 9781107152380 (hardback : alk. paper)
Subjects: LCSH: Data structures (Computer science) | Computer algorithms.
Classification: LCC QA76.9.D35 N38 2016 | DDC 005.7/3–dc23
LC record available at https://lccn.loc.gov/2016023641

ISBN 978-1-107-15238-0 Hardback

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external or third-party Internet Web sites referred to in this publication and does not guarantee that any content on such Web sites is, or will remain, accurate or appropriate.

Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

A Aylén, Facundo y Martina, que aún me creen.
A Betina, que aún me soporta.
A mi padre, a mi hermana, y a la memoria de mi madre.

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

Contents

List of Algorithms — *page* xiii
Foreword — xvii
Acknowledgments — xix

1 Introduction — **1**
- **1.1** Why Compact Data Structures? — 1
- **1.2** Why This Book? — 3
- **1.3** Organization — 4
- **1.4** Software Resources — 6
- **1.5** Mathematics and Notation — 7
- **1.6** Bibliographic Notes — 10

2 Entropy and Coding — **14**
- **2.1** Worst-Case Entropy — 14
- **2.2** Shannon Entropy — 16
- **2.3** Empirical Entropy — 17
  - **2.3.1** Bit Sequences — 18
  - **2.3.2** Sequences of Symbols — 20
- **2.4** High-Order Entropy — 21
- **2.5** Coding — 22
- **2.6** Huffman Codes — 25
  - **2.6.1** Construction — 25
  - **2.6.2** Encoding and Decoding — 26
  - **2.6.3** Canonical Huffman Codes — 27
  - **2.6.4** Better than Huffman — 30
- **2.7** Variable-Length Codes for Integers — 30
- **2.8** Jensen's Inequality — 33
- **2.9** Application: Positional Inverted Indexes — 35
- **2.10** Summary — 36
- **2.11** Bibliographic Notes — 36

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE
Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

3 Arrays **39**
3.1 Elements of Fixed Size 40
3.2 Elements of Variable Size 45
3.2.1 Sampled Pointers 46
3.2.2 Dense Pointers 47
3.3 Partial Sums 48
3.4 Applications 49
3.4.1 Constant-Time Array Initialization 49
3.4.2 Direct Access Codes 53
3.4.3 Elias-Fano Codes 57
3.4.4 Differential Encodings and Inverted Indexes 59
3.4.5 Compressed Text Collections 59
3.5 Summary 61
3.6 Bibliographic Notes 61

4 Bitvectors **64**
4.1 Access 65
4.1.1 Zero-Order Compression 65
4.1.2 High-Order Compression 71
4.2 Rank 73
4.2.1 Sparse Sampling 73
4.2.2 Constant Time 74
4.2.3 Rank on Compressed Bitvectors 76
4.3 Select 78
4.3.1 A Simple Heuristic 78
4.3.2 An $\mathcal{O}(\log \log n)$ Time Solution 80
4.3.3 Constant Time 81
4.4 Very Sparse Bitvectors 82
4.4.1 Constant-Time Select 83
4.4.2 Solving Rank 83
4.4.3 Bitvectors with Runs 86
4.5 Applications 87
4.5.1 Partial Sums Revisited 87
4.5.2 Predecessors and Successors 89
4.5.3 Dictionaries, Sets, and Hashing 91
4.6 Summary 98
4.7 Bibliographic Notes 98

5 Permutations **103**
5.1 Inverse Permutations 103
5.2 Powers of Permutations 106
5.3 Compressible Permutations 108
5.4 Applications 115
5.4.1 Two-Dimensional Points 115
5.4.2 Inverted Indexes Revisited 116
5.5 Summary 117
5.6 Bibliographic Notes 117

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

6 Sequences **120**
6.1 Using Permutations 121
6.1.1 Chunk-Level Granularity 121
6.1.2 Operations within a Chunk 123
6.1.3 Construction 126
6.1.4 Space and Time 127
6.2 Wavelet Trees 128
6.2.1 Structure 128
6.2.2 Solving Rank and Select 132
6.2.3 Construction 134
6.2.4 Compressed Wavelet Trees 136
6.2.5 Wavelet Matrices 139
6.3 Alphabet Partitioning 150
6.4 Applications 155
6.4.1 Compressible Permutations Again 155
6.4.2 Compressed Text Collections Revisited 157
6.4.3 Non-positional Inverted Indexes 157
6.4.4 Range Quantile Queries 159
6.4.5 Revisiting Arrays of Variable-Length Cells 160
6.5 Summary 161
6.6 Bibliographic Notes 162

7 Parentheses **167**
7.1 A Simple Implementation 170
7.1.1 Range Min-Max Trees 170
7.1.2 Forward and Backward Searching 175
7.1.3 Range Minima and Maxima 180
7.1.4 Rank and Select Operations 188
7.2 Improving the Complexity 188
7.2.1 Queries inside Buckets 190
7.2.2 Forward and Backward Searching 191
7.2.3 Range Minima and Maxima 196
7.2.4 Rank and Select Operations 200
7.3 Multi-Parenthesis Sequences 200
7.3.1 Nearest Marked Ancestors 201
7.4 Applications 202
7.4.1 Succinct Range Minimum Queries 202
7.4.2 XML Documents 204
7.5 Summary 207
7.6 Bibliographic Notes 207

8 Trees **211**
8.1 LOUDS: A Simple Representation 212
8.1.1 Binary and Cardinal Trees 219
8.2 Balanced Parentheses 222
8.2.1 Binary Trees Revisited 228

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

8.3 DFUDS Representation 233
8.3.1 Cardinal Trees Revisited 240
8.4 Labeled Trees 241
8.5 Applications 245
8.5.1 Routing in Minimum Spanning Trees 246
8.5.2 Grammar Compression 248
8.5.3 Tries 252
8.5.4 LZ78 Compression 259
8.5.5 XML and XPath 262
8.5.6 Treaps 264
8.5.7 Integer Functions 266
8.6 Summary 272
8.7 Bibliographic Notes 272

9 Graphs **279**
9.1 General Graphs 281
9.1.1 Using Bitvectors 281
9.1.2 Using Sequences 281
9.1.3 Undirected Graphs 284
9.1.4 Labeled Graphs 285
9.1.5 Construction 289
9.2 Clustered Graphs 291
9.2.1 K^2-Tree Structure 291
9.2.2 Queries 292
9.2.3 Reducing Space 294
9.2.4 Construction 296
9.3 K-Page Graphs 296
9.3.1 One-Page Graphs 297
9.3.2 K-Page Graphs 299
9.3.3 Construction 307
9.4 Planar Graphs 307
9.4.1 Orderly Spanning Trees 308
9.4.2 Triangulations 315
9.4.3 Construction 317
9.5 Applications 327
9.5.1 Binary Relations 327
9.5.2 RDF Datasets 328
9.5.3 Planar Routing 330
9.5.4 Planar Drawings 336
9.6 Summary 338
9.7 Bibliographic Notes 338

10 Grids **347**
10.1 Wavelet Trees 348
10.1.1 Counting 350
10.1.2 Reporting 353
10.1.3 Sorted Reporting 355

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

10.2 K^2-Trees 357
10.2.1 Reporting 359
10.3 Weighted Points 362
10.3.1 Wavelet Trees 362
10.3.2 K^2-Trees 365
10.4 Higher Dimensions 371
10.5 Applications 372
10.5.1 Dominating Points 372
10.5.2 Geographic Information Systems 373
10.5.3 Object Visibility 377
10.5.4 Position-Restricted Searches on Suffix Arrays 379
10.5.5 Searching for Fuzzy Patterns 380
10.5.6 Indexed Searching in Grammar-Compressed Text 382
10.6 Summary 388
10.7 Bibliographic Notes 388

11 Texts **395**
11.1 Compressed Suffix Arrays 397
11.1.1 Replacing A with Ψ 398
11.1.2 Compressing Ψ 399
11.1.3 Backward Search 401
11.1.4 Locating and Displaying 403
11.2 The FM-Index 406
11.3 High-Order Compression 409
11.3.1 The Burrows-Wheeler Transform 409
11.3.2 High-Order Entropy 410
11.3.3 Partitioning L into Uniform Chunks 413
11.3.4 High-Order Compression of Ψ 414
11.4 Construction 415
11.4.1 Suffix Array Construction 415
11.4.2 Building the BWT 416
11.4.3 Building Ψ 418
11.5 Suffix Trees 419
11.5.1 Longest Common Prefixes 419
11.5.2 Suffix Tree Operations 420
11.5.3 A Compact Representation 424
11.5.4 Construction 426
11.6 Applications 429
11.6.1 Finding Maximal Substrings of a Pattern 429
11.6.2 Labeled Trees Revisited 432
11.6.3 Document Retrieval 438
11.6.4 XML Retrieval Revisited 441
11.7 Summary 442
11.8 Bibliographic Notes 442

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

12 Dynamic Structures **450**
12.1 Bitvectors 450
12.1.1 Solving Queries 452
12.1.2 Handling Updates 452
12.1.3 Compressed Bitvectors 461
12.2 Arrays and Partial Sums 463
12.3 Sequences 465
12.4 Trees 467
12.4.1 LOUDS Representation 469
12.4.2 BP Representation 472
12.4.3 DFUDS Representation 474
12.4.4 Dynamic Range Min-Max Trees 476
12.4.5 Labeled Trees 479
12.5 Graphs and Grids 480
12.5.1 Dynamic Wavelet Matrices 480
12.5.2 Dynamic k^2-Trees 482
12.6 Texts 485
12.6.1 Insertions 485
12.6.2 Document Identifiers 486
12.6.3 Samplings 486
12.6.4 Deletions 490
12.7 Memory Allocation 492
12.8 Summary 494
12.9 Bibliographic Notes 494

13 Recent Trends **501**
13.1 Encoding Data Structures 502
13.1.1 Effective Entropy 502
13.1.2 The Entropy of RMQs 503
13.1.3 Expected Effective Entropy 504
13.1.4 Other Encoding Problems 504
13.2 Repetitive Text Collections 508
13.2.1 Lempel-Ziv Compression 509
13.2.2 Lempel-Ziv Indexing 513
13.2.3 Faster and Larger Indexes 516
13.2.4 Compressed Suffix Arrays and Trees 519
13.3 Secondary Memory 523
13.3.1 Bitvectors 524
13.3.2 Sequences 527
13.3.3 Trees 528
13.3.4 Grids and Graphs 530
13.3.5 Texts 534

Index 549

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

List of Algorithms

2.1	Building a prefix code given the desired lengths	*page* 24
2.2	Building a Huffman tree	27
2.3	Building a Canonical Huffman code representation	29
2.4	Reading a symbol with a Canonical Huffman code	29
2.5	Various integer encodings	34
3.1	Reading and writing on bit arrays	41
3.2	Reading and writing on fixed-length cell arrays	44
3.3	Manipulating initializable arrays	52
3.4	Reading from a direct access code representation	55
3.5	Creating direct access codes from an array	56
3.6	Finding optimal piece lengths for direct access codes	58
3.7	Intersection of inverted lists	60
4.1	Encoding and decoding bit blocks as pairs (c, o)	67
4.2	Answering access on compressed bitvectors	69
4.3	Answering rank with sparse sampling	74
4.4	Answering rank with dense sampling	75
4.5	Answering rank on compressed bitvectors	77
4.6	Answering select with sparse sampling	80
4.7	Building the select structures	82
4.8	Answering select and rank on very sparse bitvectors	85
4.9	Building the structures for very sparse bitvectors	86
4.10	Building a perfect hash function	94
5.1	Answering π^{-1} with shortcuts	105
5.2	Building the shortcut structure	107
5.3	Answering π^k with the cycle decomposition	108
5.4	Answering π and π^{-1} on compressible permutations	112
5.5	Building the compressed permutation representation, part 1	113
5.6	Building the compressed permutation representation, part 2	114
6.1	Answering queries with the permutation-based structure	125
6.2	Building the permutation-based representation of a sequence	126

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

6.3 Answering access and rank with wavelet trees 131
6.4 Answering select with wavelet trees 134
6.5 Building a wavelet tree 135
6.6 Answering access and rank with wavelet matrices 143
6.7 Answering select with wavelet matrices 144
6.8 Building a wavelet matrix 145
6.9 Building a suitable Huffman code for wavelet matrices 149
6.10 Building a wavelet matrix from Huffman codes 150
6.11 Answering queries with alphabet partitioning 153
6.12 Building the alphabet partitioning representation 155
6.13 Answering π and π^{-1} using sequences 156
6.14 Inverted list intersection using a sequence representation 158
6.15 Non-positional inverted list intersection 159
6.16 Solving range quantile queries on wavelet trees 161
7.1 Converting between leaf numbers and positions of rmM-trees 171
7.2 Building the C table for the rmM-trees 174
7.3 Building the rmM-tree 175
7.4 Scanning a block for fwdsearch(i, d) 177
7.5 Computing fwdsearch(i, d) 178
7.6 Computing bwdsearch(i, d) 181
7.7 Scanning a block for min(i, j) 182
7.8 Computing the minimum excess in $B[i, j]$ 183
7.9 Computing mincount(i, j) 186
7.10 Computing minselect(i, j, t) 187
7.11 Computing rank$_{10}(i)$ on B 189
7.12 Computing select$_{10}(j)$ on B 189
7.13 Finding the smallest segment of a type containing a position 202
7.14 Solving rmq$_A$ with $2n$ parentheses 204
7.15 Building the structure for succinct RMQs 205
8.1 Computing the ordinal tree operations using LOUDS 216
8.2 Computing lca(u, v) on the LOUDS representation 217
8.3 Building the LOUDS representation 218
8.4 Computing the cardinal tree operations using LOUDS 220
8.5 Computing basic binary tree operations using LOUDS 221
8.6 Building the BP representation of an ordinal tree 223
8.7 Computing the simple BP operations on ordinal trees 225
8.8 Computing the complex BP operations on ordinal trees 227
8.9 Building the BP representation of a binary tree 230
8.10 Computing basic binary tree operations using BP 231
8.11 Computing advanced binary tree operations using BP 234
8.12 Building the DFUDS representation 235
8.13 Computing the simple DFUDS operations on ordinal trees 239
8.14 Computing the complex DFUDS operations on ordinal trees 240
8.15 Computing the additional cardinal tree operations on DFUDS 241
8.16 Computing the labeled tree operations on LOUDS or DFUDS 244
8.17 Enumerating the path from u to v with LOUDS 247

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

8.18 Extraction and pattern search in tries 255
8.19 Extraction of a text substring from its LZ78 representation 262
8.20 Reporting the largest values in a range using a treap 265
8.21 Computing $f^k(i)$ with the compact representation 268
8.22 Computing $f^{-k}(i)$ with the compact representation 269
9.1 Operations on general directed graphs 283
9.2 Operations on general undirected graphs 284
9.3 Operations on labeled directed graphs 289
9.4 Label-specific operations on directed graphs 290
9.5 Operation adj on a k^2-tree 293
9.6 Operations neigh and rneigh on a k^2-tree 294
9.7 Building the k^2-tree 297
9.8 Operations on one-page graphs 300
9.9 Operations degree and neigh on k-page graphs 304
9.10 Operation adj on k-page graphs 305
9.11 Operations on planar graphs 312
9.12 Finding which neighbor of u is v on planar graphs 313
9.13 Additional operations on the planar graph representation 314
9.14 Operations neigh and degree on triangular graphs 317
9.15 Operation adj on triangular graphs 318
9.16 Object-object join on RDF graphs using k^2-trees 331
9.17 Subject-object join on RDF graphs using k^2-trees 332
9.18 Routing on a planar graph through locally maximum benefit 333
9.19 Routing on a planar graph through face traversals 334
9.20 Two-visibility drawing of a planar graph 337
10.1 Answering count with a wavelet matrix 351
10.2 Procedures for report on a wavelet matrix 354
10.3 Finding the leftmost point in a range with a wavelet matrix 356
10.4 Finding the highest points in a range with a wavelet matrix 357
10.5 Procedure for report on a k^2-tree 360
10.6 Answering top with a wavelet matrix 363
10.7 Prioritized traversal for top on a k^2-tree 368
10.8 Recursive traversal for top on a k^2-tree 370
10.9 Procedure for closest on a k^2-tree 375
10.10 Searching for P in a grammar-compressed text T 387
11.1 Comparing P with $T[A[i], n]$ using Ψ 399
11.2 Backward search on a compressed suffix array 402
11.3 Obtaining $A[i]$ on a compressed suffix array 404
11.4 Displaying $T[j, j+\ell-1]$ on a compressed suffix array 405
11.5 Backward search on an FM-index 406
11.6 Obtaining $A[i]$ on an FM-index 408
11.7 Displaying $T[j, j+\ell-1]$ on an FM-index 408
11.8 Building the BWT of a text T in compact space 417
11.9 Generating the partition of A for BWT construction 418
11.10 Computing the suffix tree operations 425
11.11 Building the suffix tree components 429

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

11.12 Finding the maximal intervals of P that occur often in T 431
11.13 Emulating operations on virtual suffix tree nodes 433
11.14 Subpath search on BWT-like encoded labeled trees 435
11.15 Navigation on BWT-like encoded labeled trees 437
11.16 Document listing 439
12.1 Answering access and rank queries on a dynamic bitvector 453
12.2 Answering select queries on a dynamic bitvector 454
12.3 Processing insert on a dynamic bitvector 456
12.4 Processing delete on a dynamic bitvector, part 1 458
12.5 Processing delete on a dynamic bitvector, part 2 459
12.6 Processing bitset and bitclear on a dynamic bitvector 460
12.7 Answering access queries on a sparse dynamic bitvector 463
12.8 Inserting and deleting symbols on a dynamic wavelet tree 466
12.9 Inserting and deleting symbols on a dynamic wavelet matrix 468
12.10 Inserting and deleting leaves in a LOUDS representation 470
12.11 Inserting and deleting leaves in a LOUDS cardinal tree 471
12.12 Inserting and deleting nodes in a BP representation 473
12.13 Inserting and deleting nodes in a DFUDS representation 475
12.14 Inserting parentheses on a dynamic rmM-tree 477
12.15 Computing fwdsearch(i, d) on a dynamic rmM-tree 478
12.16 Computing the minimum excess in a dynamic rmM-tree 479
12.17 Inserting and deleting grid points using a wavelet matrix 481
12.18 Inserting and deleting grid points using a k^2-tree 483
12.19 Inserting a document on a dynamic FM-index 488
12.20 Locating and displaying on a dynamic FM-index 489
12.21 Deleting a document on a dynamic FM-index 491
13.1 Reporting τ-majorities from an encoding 508
13.2 Performing the LZ76 parsing 512
13.3 Reporting occurrences on the LZ76-index 517
13.4 Answering count with a wavelet matrix on disk 531
13.5 Backward search on a reduced FM-index 538

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

Foreword

This is a delightful book on data structures that are both time and space efficient. Space as well as time efficiency is crucial in modern information systems. Even if we have extra space somewhere, it is unlikely to be close to the processors. The space used by most such systems is overwhelmingly for structural indexing, such as B-trees, hash tables, and various cross-references, rather than for "raw data." Indeed data, such as text, take far too much space in raw form and must be compressed. A system that keeps both data and indices in a compact form has a major advantage.

Hence the title of the book. Gonzalo Navarro uses the term "compact data structures" to describe a newly emerging research area. It has developed from two distinct but interrelated topics. The older is that of text compression, dating back to the work of Shannon, Fano, and Huffman (among others) in the late 1940s and early 1950s (although text compression as such was not their main concern). Through the last half of the 20th century, as the size of the text to be processed increased and computing platforms became more powerful, algorithmics and information theory became much more sophisticated. The goal of data compression, at least until the year 2000 or so, simply meant compressing information as well as possible and then decompressing each time it was needed. A hallmark of compact data structures is working with text in compressed form saving both decompression time and space. The newer contributing area evolved in the 1990s after the work of Jacobson and is generally referred to as "succinct data structures." The idea is to represent a combinatorial object, such as a graph, tree, or sparse bit vector, in a number of bits that differs from the information theory lower bound by only a lower order term. So, for example, a binary tree on n nodes takes only $2n + o(n)$ bits. The trick is to perform the necessary operations, e.g., find child, parent, or subtree size, in constant time.

Compact data structures take into account both "data" and "structures" and are a little more tolerant of "best effort" than one might be with exact details of information theoretic lower bounds. Here the subtitle, "A Practical Approach," comes into play. The emphasis is on methods that are reasonable to implement and appropriate for today's (and tomorrow's) data sizes, rather than on the asymptotics that one sees with the "theoretical approach."

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

Reading the book, I was taken with the thorough coverage of the topic and the clarity of presentation. Finding, easily, specific results was, well, easy, as suits the experienced researcher in the field. On the other hand, the careful exposition of key concepts, with elucidating examples, makes it ideal as a graduate text or for the researcher from a tangentially related area. The book covers the historical and mathematical background along with the key developments of the 1990s and early years of the current century, which form its core. Text indexing has been a major driving force for the area, and techniques for it are nicely covered. The final two chapters point to long-term challenges and recent advances. Updates to compact data structures have been a problem for as long as the topic has been studied. The treatment here is not only state of the art but will undoubtedly be a major influence on further improvements to dynamic structures, a key aspect of improving their applicability. The final chapter focuses on encodings, working with repetitive text, and issues of the memory hierarchy. The book will be a key reference and guiding light in the field for years to come.

J. Ian Munro
University of Waterloo

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-15238-0 — Compact Data Structures
Gonzalo Navarro
Frontmatter
More Information

Acknowledgments

I am indebted to Joshimar Córdova and Simon Gog, who took the time to exhaustively read large portions of the book. They made a number of useful comments and killed many dangerous bugs. Several other students and colleagues read parts of the book and also made useful suggestions: Travis Gagie, Patricio Huepe, Roberto Konow, Susana Ladra, Veli Mäkinen, Miguel Ángel Martínez-Prieto, Ian Munro, and Alberto Ordóñez. Others, like Yakov Nekrich, Rajeev Raman, and Kunihiko Sadakane, saved me hours of searching by providing instant answers to my questions. Last but not least, Renato Cerro carefully polished my English grammar. It is most likely that some bugs remain, for which I am the only one to blame.

Ian Munro enthusiastically agreed to write the Foreword of the book. My thanks, again, to a pioneer of this beautiful area.

I would also like to thank my family for bearing with me along this two-year-long effort. It has been much more fun for me than for them.

Finally, I wish to thank the Department of Computer Science at the University of Chile for giving me the opportunity of a life dedicated to academia in a friendly and supportive environment.

© in this web service Cambridge University Press www.cambridge.org