

# Contents

---

<i>List of Algorithms</i>	<i>page</i> xiii
<i>Foreword</i>	xvii
<i>Acknowledgments</i>	xix
<b>1 Introduction</b>	<b>1</b>
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
<b>2 Entropy and Coding</b>	<b>14</b>
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

viii	CONTENTS	
<b>3</b>	<b>Arrays</b>	<b>39</b>
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
<b>4</b>	<b>Bitvectors</b>	<b>64</b>
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
<b>5</b>	<b>Permutations</b>	<b>103</b>
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

## CONTENTS

ix

<b>6 Sequences</b>	<b>120</b>
<b>6.1</b> Using Permutations	121
<b>6.1.1</b> Chunk-Level Granularity	121
<b>6.1.2</b> Operations within a Chunk	123
<b>6.1.3</b> Construction	126
<b>6.1.4</b> Space and Time	127
<b>6.2</b> Wavelet Trees	128
<b>6.2.1</b> Structure	128
<b>6.2.2</b> Solving Rank and Select	132
<b>6.2.3</b> Construction	134
<b>6.2.4</b> Compressed Wavelet Trees	136
<b>6.2.5</b> Wavelet Matrices	139
<b>6.3</b> Alphabet Partitioning	150
<b>6.4</b> Applications	155
<b>6.4.1</b> Compressible Permutations Again	155
<b>6.4.2</b> Compressed Text Collections Revisited	157
<b>6.4.3</b> Non-positional Inverted Indexes	157
<b>6.4.4</b> Range Quantile Queries	159
<b>6.4.5</b> Revisiting Arrays of Variable-Length Cells	160
<b>6.5</b> Summary	161
<b>6.6</b> Bibliographic Notes	162
<b>7 Parentheses</b>	<b>167</b>
<b>7.1</b> A Simple Implementation	170
<b>7.1.1</b> Range Min-Max Trees	170
<b>7.1.2</b> Forward and Backward Searching	175
<b>7.1.3</b> Range Minima and Maxima	180
<b>7.1.4</b> Rank and Select Operations	188
<b>7.2</b> Improving the Complexity	188
<b>7.2.1</b> Queries inside Buckets	190
<b>7.2.2</b> Forward and Backward Searching	191
<b>7.2.3</b> Range Minima and Maxima	196
<b>7.2.4</b> Rank and Select Operations	200
<b>7.3</b> Multi-Parenthesis Sequences	200
<b>7.3.1</b> Nearest Marked Ancestors	201
<b>7.4</b> Applications	202
<b>7.4.1</b> Succinct Range Minimum Queries	202
<b>7.4.2</b> XML Documents	204
<b>7.5</b> Summary	207
<b>7.6</b> Bibliographic Notes	207
<b>8 Trees</b>	<b>211</b>
<b>8.1</b> LOUDS: A Simple Representation	212
<b>8.1.1</b> Binary and Cardinal Trees	219
<b>8.2</b> Balanced Parentheses	222
<b>8.2.1</b> Binary Trees Revisited	228

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
<b>9</b>	<b>Graphs</b>	<b>279</b>
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
<b>10</b>	<b>Grids</b>	<b>347</b>
10.1	Wavelet Trees	348
10.1.1	Counting	350
10.1.2	Reporting	353
10.1.3	Sorted Reporting	355

## CONTENTS

xi

<b>10.2</b>	$K^2$ -Trees	357
	<b>10.2.1</b> Reporting	359
<b>10.3</b>	Weighted Points	362
	<b>10.3.1</b> Wavelet Trees	362
	<b>10.3.2</b> $K^2$ -Trees	365
<b>10.4</b>	Higher Dimensions	371
<b>10.5</b>	Applications	372
	<b>10.5.1</b> Dominating Points	372
	<b>10.5.2</b> Geographic Information Systems	373
	<b>10.5.3</b> Object Visibility	377
	<b>10.5.4</b> Position-Restricted Searches on Suffix Arrays	379
	<b>10.5.5</b> Searching for Fuzzy Patterns	380
	<b>10.5.6</b> Indexed Searching in Grammar-Compressed Text	382
<b>10.6</b>	Summary	388
<b>10.7</b>	Bibliographic Notes	388
<b>11</b>	<b>Texts</b>	<b>395</b>
<b>11.1</b>	Compressed Suffix Arrays	397
	<b>11.1.1</b> Replacing $A$ with $\Psi$	398
	<b>11.1.2</b> Compressing $\Psi$	399
	<b>11.1.3</b> Backward Search	401
	<b>11.1.4</b> Locating and Displaying	403
<b>11.2</b>	The FM-Index	406
<b>11.3</b>	High-Order Compression	409
	<b>11.3.1</b> The Burrows-Wheeler Transform	409
	<b>11.3.2</b> High-Order Entropy	410
	<b>11.3.3</b> Partitioning $L$ into Uniform Chunks	413
	<b>11.3.4</b> High-Order Compression of $\Psi$	414
<b>11.4</b>	Construction	415
	<b>11.4.1</b> Suffix Array Construction	415
	<b>11.4.2</b> Building the BWT	416
	<b>11.4.3</b> Building $\Psi$	418
<b>11.5</b>	Suffix Trees	419
	<b>11.5.1</b> Longest Common Prefixes	419
	<b>11.5.2</b> Suffix Tree Operations	420
	<b>11.5.3</b> A Compact Representation	424
	<b>11.5.4</b> Construction	426
<b>11.6</b>	Applications	429
	<b>11.6.1</b> Finding Maximal Substrings of a Pattern	429
	<b>11.6.2</b> Labeled Trees Revisited	432
	<b>11.6.3</b> Document Retrieval	438
	<b>11.6.4</b> XML Retrieval Revisited	441
<b>11.7</b>	Summary	442
<b>11.8</b>	Bibliographic Notes	442

<b>xii</b>	<b>CONTENTS</b>	
<b>12</b>	<b>Dynamic Structures</b>	<b>450</b>
<b>12.1</b>	Bitvectors	450
<b>12.1.1</b>	Solving Queries	452
<b>12.1.2</b>	Handling Updates	452
<b>12.1.3</b>	Compressed Bitvectors	461
<b>12.2</b>	Arrays and Partial Sums	463
<b>12.3</b>	Sequences	465
<b>12.4</b>	Trees	467
<b>12.4.1</b>	LOUDS Representation	469
<b>12.4.2</b>	BP Representation	472
<b>12.4.3</b>	DFUDS Representation	474
<b>12.4.4</b>	Dynamic Range Min-Max Trees	476
<b>12.4.5</b>	Labeled Trees	479
<b>12.5</b>	Graphs and Grids	480
<b>12.5.1</b>	Dynamic Wavelet Matrices	480
<b>12.5.2</b>	Dynamic $k^2$ -Trees	482
<b>12.6</b>	Texts	485
<b>12.6.1</b>	Insertions	485
<b>12.6.2</b>	Document Identifiers	486
<b>12.6.3</b>	Samplings	486
<b>12.6.4</b>	Deletions	490
<b>12.7</b>	Memory Allocation	492
<b>12.8</b>	Summary	494
<b>12.9</b>	Bibliographic Notes	494
<b>13</b>	<b>Recent Trends</b>	<b>501</b>
<b>13.1</b>	Encoding Data Structures	502
<b>13.1.1</b>	Effective Entropy	502
<b>13.1.2</b>	The Entropy of RMQs	503
<b>13.1.3</b>	Expected Effective Entropy	504
<b>13.1.4</b>	Other Encoding Problems	504
<b>13.2</b>	Repetitive Text Collections	508
<b>13.2.1</b>	Lempel-Ziv Compression	509
<b>13.2.2</b>	Lempel-Ziv Indexing	513
<b>13.2.3</b>	Faster and Larger Indexes	516
<b>13.2.4</b>	Compressed Suffix Arrays and Trees	519
<b>13.3</b>	Secondary Memory	523
<b>13.3.1</b>	Bitvectors	524
<b>13.3.2</b>	Sequences	527
<b>13.3.3</b>	Trees	528
<b>13.3.4</b>	Grids and Graphs	530
<b>13.3.5</b>	Texts	534
	<i>Index</i>	549

## List of Algorithms

---

2.1 Building a prefix code given the desired lengths	<i>page 24</i>
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 <b>access</b> on compressed bitvectors	69
4.3 Answering <b>rank</b> with sparse sampling	74
4.4 Answering <b>rank</b> with dense sampling	75
4.5 Answering <b>rank</b> on compressed bitvectors	77
4.6 Answering <b>select</b> with sparse sampling	80
4.7 Building the <b>select</b> structures	82
4.8 Answering <b>select</b> and <b>rank</b> 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 $\pi^{-1}$ with shortcuts	105
5.2 Building the shortcut structure	107
5.3 Answering $\pi^k$ with the cycle decomposition	108
5.4 Answering $\pi$ and $\pi^{-1}$ 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

6.3	Answering <b>access</b> and <b>rank</b> with wavelet trees	131
6.4	Answering <b>select</b> with wavelet trees	134
6.5	Building a wavelet tree	135
6.6	Answering <b>access</b> and <b>rank</b> with wavelet matrices	143
6.7	Answering <b>select</b> 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 $\pi$ and $\pi^{-1}$ 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 $\text{fwdsearch}(i, d)$	177
7.5	Computing $\text{fwdsearch}(i, d)$	178
7.6	Computing $\text{bwdsearch}(i, d)$	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 $\text{mincount}(i, j)$	186
7.10	Computing $\text{minselect}(i, j, t)$	187
7.11	Computing $\text{rank}_{10}(i)$ on $B$	189
7.12	Computing $\text{select}_{10}(j)$ on $B$	189
7.13	Finding the smallest segment of a type containing a position	202
7.14	Solving $\text{rmq}_A$ 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 $\text{lca}(u, v)$ 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



## LIST OF ALGORITHMS

xv

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 <code>adj</code> on a $k^2$ -tree	293
9.6	Operations <code>neigh</code> and <code>rneigh</code> 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 <code>degree</code> and <code>neigh</code> on $k$ -page graphs	304
9.10	Operation <code>adj</code> 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 <code>neigh</code> and <code>degree</code> on triangular graphs	317
9.15	Operation <code>adj</code> 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 <code>count</code> with a wavelet matrix	351
10.2	Procedures for <code>report</code> 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 <code>report</code> on a $k^2$ -tree	360
10.6	Answering <code>top</code> with a wavelet matrix	363
10.7	Prioritized traversal for <code>top</code> on a $k^2$ -tree	368
10.8	Recursive traversal for <code>top</code> on a $k^2$ -tree	370
10.9	Procedure for <code>closest</code> 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 $\Psi$	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

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 <b>access</b> and <b>rank</b> queries on a dynamic bitvector	453
12.2	Answering <b>select</b> queries on a dynamic bitvector	454
12.3	Processing <b>insert</b> on a dynamic bitvector	456
12.4	Processing <b>delete</b> on a dynamic bitvector, part 1	458
12.5	Processing <b>delete</b> on a dynamic bitvector, part 2	459
12.6	Processing <b>bitset</b> and <b>bitclear</b> on a dynamic bitvector	460
12.7	Answering <b>access</b> 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 <b> fwdsearch</b> ( $i, d$ ) 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 $\tau$ -majorities from an encoding	508
13.2	Performing the LZ76 parsing	512
13.3	Reporting occurrences on the LZ76-index	517
13.4	Answering <b>count</b> with a wavelet matrix on disk	531
13.5	Backward search on a reduced FM-index	538