Contents

List of Algorithms 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
# CONTENTS

## 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 \(O(\log \log n)\) 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
## Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Sequences</td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>Using Permutations</td>
<td>120</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Chunk-Level Granularity</td>
<td>121</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Operations within a Chunk</td>
<td>122</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Construction</td>
<td>124</td>
</tr>
<tr>
<td>6.1.4</td>
<td>Space and Time</td>
<td>126</td>
</tr>
<tr>
<td>6.2</td>
<td>Wavelet Trees</td>
<td>121</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Structure</td>
<td>128</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Solving Rank and Select</td>
<td>130</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Construction</td>
<td>132</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Compressed Wavelet Trees</td>
<td>134</td>
</tr>
<tr>
<td>6.2.5</td>
<td>Wavelet Matrices</td>
<td>136</td>
</tr>
<tr>
<td>6.3</td>
<td>Alphabet Partitioning</td>
<td>140</td>
</tr>
<tr>
<td>6.4</td>
<td>Applications</td>
<td>150</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Compressible Permutations Again</td>
<td>155</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Compressed Text Collections Revisited</td>
<td>157</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Non-positional Inverted Indexes</td>
<td>159</td>
</tr>
<tr>
<td>6.4.4</td>
<td>Range Quantile Queries</td>
<td>160</td>
</tr>
<tr>
<td>6.4.5</td>
<td>Revisiting Arrays of Variable-Length Cells</td>
<td>162</td>
</tr>
<tr>
<td>6.5</td>
<td>Summary</td>
<td>166</td>
</tr>
<tr>
<td>6.6</td>
<td>Bibliographic Notes</td>
<td>162</td>
</tr>
<tr>
<td>7</td>
<td>Parentheses</td>
<td></td>
</tr>
<tr>
<td>7.1</td>
<td>A Simple Implementation</td>
<td>167</td>
</tr>
<tr>
<td>7.1.1</td>
<td>Range Min-Max Trees</td>
<td>170</td>
</tr>
<tr>
<td>7.1.2</td>
<td>Forward and Backward Searching</td>
<td>175</td>
</tr>
<tr>
<td>7.1.3</td>
<td>Range Minima and Maxima</td>
<td>180</td>
</tr>
<tr>
<td>7.1.4</td>
<td>Rank and Select Operations</td>
<td>188</td>
</tr>
<tr>
<td>7.2</td>
<td>Improving the Complexity</td>
<td>188</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Queries inside Buckets</td>
<td>190</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Forward and Backward Searching</td>
<td>191</td>
</tr>
<tr>
<td>7.2.3</td>
<td>Range Minima and Maxima</td>
<td>196</td>
</tr>
<tr>
<td>7.2.4</td>
<td>Rank and Select Operations</td>
<td>200</td>
</tr>
<tr>
<td>7.3</td>
<td>Multi-Parenthesis Sequences</td>
<td>200</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Nearest Marked Ancestors</td>
<td>205</td>
</tr>
<tr>
<td>7.4</td>
<td>Applications</td>
<td>206</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Succinct Range Minimum Queries</td>
<td>207</td>
</tr>
<tr>
<td>7.4.2</td>
<td>XML Documents</td>
<td>208</td>
</tr>
<tr>
<td>7.5</td>
<td>Summary</td>
<td>208</td>
</tr>
<tr>
<td>7.6</td>
<td>Bibliographic Notes</td>
<td>209</td>
</tr>
<tr>
<td>8</td>
<td>Trees</td>
<td></td>
</tr>
<tr>
<td>8.1</td>
<td>LOUDS: A Simple Representation</td>
<td>211</td>
</tr>
<tr>
<td>8.1.1</td>
<td>Binary and Cardinal Trees</td>
<td>212</td>
</tr>
<tr>
<td>8.2</td>
<td>Balanced Parentheses</td>
<td>222</td>
</tr>
<tr>
<td>8.2.1</td>
<td>Binary Trees Revisited</td>
<td>228</td>
</tr>
</tbody>
</table>
## CONTENTS

### 8.3 DFUDS Representation
- 8.3.1 Cardinal Trees Revisited

### 8.4 Labeled Trees

### 8.5 Applications
- 8.5.1 Routing in Minimum Spanning Trees
- 8.5.2 Grammar Compression
- 8.5.3 Tries
- 8.5.4 LZ78 Compression
- 8.5.5 XML and XPath
- 8.5.6 Treaps
- 8.5.7 Integer Functions

### 8.6 Summary

### 8.7 Bibliographic Notes

### 9 Graphs

#### 9.1 General Graphs
- 9.1.1 Using Bitvectors
- 9.1.2 Using Sequences
- 9.1.3 Undirected Graphs
- 9.1.4 Labeled Graphs
- 9.1.5 Construction

#### 9.2 Clustered Graphs
- 9.2.1 $K^2$-Tree Structure
- 9.2.2 Queries
- 9.2.3 Reducing Space
- 9.2.4 Construction

#### 9.3 $K$-Page Graphs
- 9.3.1 One-Page Graphs
- 9.3.2 $K$-Page Graphs
- 9.3.3 Construction

#### 9.4 Planar Graphs
- 9.4.1 Orderly Spanning Trees
- 9.4.2 Triangulations
- 9.4.3 Construction

#### 9.5 Applications
- 9.5.1 Binary Relations
- 9.5.2 RDF Datasets
- 9.5.3 Planar Routing
- 9.5.4 Planar Drawings

### 9.6 Summary

### 9.7 Bibliographic Notes

### 10 Grids

#### 10.1 Wavelet Trees
- 10.1.1 Counting
- 10.1.2 Reporting
- 10.1.3 Sorted Reporting

© in this web service Cambridge University Press

www.cambridge.org
## CONTENTS

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

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

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
# List of Algorithms

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

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 \( \pi \) 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 `fwdsearch`(\( i, d \)) 177  
7.5 Computing `fwdsearch`(\( i, d \)) 178  
7.6 Computing `bwdsearch`(\( i, d \)) 181  
7.7 Scanning a block for \( \text{min}(i, j) \) 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`(\( i, j, t \)) on \( B \) 189  
7.12 Computing `select`(\( i, j, t \)) on \( B \) 189  
7.13 Finding the smallest segment of a type containing a position 202  
7.14 Solving \( \text{rmq}_A \) with \( 2^n \) 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
LIST OF ALGORITHMS

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 \( \text{adj} \) on a \( k^2 \)-tree 293
9.6 Operations \( \text{neigh} \) and \( \text{rneigh} \) 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 \( \text{degree} \) and \( \text{neigh} \) on \( k \)-page graphs 304
9.10 Operation \( \text{adj} \) 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 \( \text{neigh} \) and \( \text{degree} \) on triangular graphs 317
9.15 Operation \( \text{adj} \) 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 \( \text{count} \) with a wavelet matrix 351
10.2 Procedures for \( \text{report} \) 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 \( \text{report} \) on a \( k^2 \)-tree 360
10.6 Answering \( \text{top} \) with a wavelet matrix 363
10.7 Prioritized traversal for \( \text{top} \) on a \( k^2 \)-tree 368
10.8 Recursive traversal for \( \text{top} \) on a \( k^2 \)-tree 370
10.9 Procedure for \( \text{closest} \) 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 \( \Psi \) 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
## LIST OF ALGORITHMS

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 r-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