

125 Problems in Text Algorithms

String matching is one of the oldest algorithmic techniques, yet still one of the most pervasive in computer science. The past 20 years have seen technological leaps in applications as diverse as information retrieval and compression. This copiously illustrated collection of puzzles and exercises in key areas of text algorithms and combinatorics on words offers graduate students and researchers a pleasant and direct way to learn and practice with advanced concepts.

The problems are drawn from a large range of scientific publications, both classic and new. Building up from the basics, the book goes on to showcase problems in combinatorics on words (including Fibonacci or Thue–Morse words), pattern matching (including Knuth–Morris–Pratt and Boyer–Moore–like algorithms), efficient text data structures (including suffix trees and suffix arrays), regularities in words (including periods and runs) and text compression (including Huffman, Lempel–Ziv and Burrows–Wheeler–based methods).

MAXIME CROCHEMORE is Emeritus Professor at Université Gustave Eiffel and of King’s College London. He holds an honorary doctorate from the University of Helsinki. He is the author of more than 200 articles on algorithms on strings and their applications, and co-author of several books on the subject.

THIERRY LECROQ is a professor in the Department of Computer Science at the University of Rouen Normandy (France). He is currently head of the research team Information Processing in Biology and Health of the Laboratory of Computer Science, Information Processing and System. He has been one of the coordinators of the working group in stringology of the French National Centre for Scientific Research for more than 10 years.

WOJCIECH RYTTER is a professor at the Faculty of Mathematics, Informatics and Mechanics, University of Warsaw. He is the author of a large number of publications on automata, formal languages, parallel algorithms and algorithms on texts. He is a co-author of several books on these subjects, including *Efficient Parallel Algorithms*, *Text Algorithms* and *Analysis of Algorithms and Data Structures*. He is a member of Academia Europaea.

125 Problems in Text Algorithms

With Solutions

MAXIME CROCHEMORE

Gustave Eiffel University

THIERRY LECROQ

University of Rouen Normandy

WOJCIECH RYTTER

University of Warsaw



CAMBRIDGE
UNIVERSITY PRESS

www.cambridge.org

Contents

<i>Preface</i>	<i>page ix</i>
1 The Very Basics of Stringology	1
2 Combinatorial Puzzles	17
1 Stringologic Proof of Fermat’s Little Theorem	18
2 Simple Case of Codicity Testing	19
3 Magic Squares and the Thue–Morse Word	20
4 Oldenburger–Kolakoski Sequence	22
5 Square-Free Game	24
6 Fibonacci Words and Fibonacci Numeration System	26
7 Wythoff’s Game and Fibonacci Word	28
8 Distinct Periodic Words	30
9 A Relative of the Thue–Morse Word	33
10 Thue–Morse Words and Sums of Powers	34
11 Conjugates and Rotations of Words	35
12 Conjugate Palindromes	37
13 Many Words with Many Palindromes	39
14 Short Superword of Permutations	41
15 Short Supersequence of Permutations	43
16 Skolem Words	45
17 Langford Words	48
18 From Lyndon Words to de Bruijn Words	50
3 Pattern Matching	53
19 Border Table	54
20 Shortest Covers	56
21 Short Borders	58

22	Prefix Table	60
23	Border Table to the Maximal Suffix	62
24	Periodicity Test	65
25	Strict Borders	67
26	Delay of Sequential String Matching	70
27	Sparse Matching Automaton	72
28	Comparison-Effective String Matching	74
29	Strict Border Table of the Fibonacci Word	76
30	Words with Singleton Variables	78
31	Order-Preserving Patterns	81
32	Parameterised Matching	83
33	Good-Suffix Table	85
34	Worst Case of the Boyer–Moore Algorithm	88
35	Turbo-BM Algorithm	90
36	String Matching with Don't Cares	92
37	Cyclic Equivalence	93
38	Simple Maximal Suffix Computation	96
39	Self-Maximal Words	98
40	Maximal Suffix and Its Period	100
41	Critical Position of a Word	103
42	Periods of Lyndon Word Prefixes	105
43	Searching Zimin Words	107
44	Searching Irregular 2D Patterns	110
4	Efficient Data Structures	111
45	List Algorithm for Shortest Cover	112
46	Computing Longest Common Prefixes	113
47	Suffix Array to Suffix Tree	115
48	Linear Suffix Trie	119
49	Ternary Search Trie	122
50	Longest Common Factor of Two Words	124
51	Subsequence Automaton	126
52	Codicity Test	128
53	LPF Table	130
54	Sorting Suffixes of Thue–Morse Words	134
55	Bare Suffix Tree	137
56	Comparing Suffixes of a Fibonacci Word	139
57	Avoidability of Binary Words	141
58	Avoiding a Set of Words	144

59	Minimal Unique Factors	146
60	Minimal Absent Words	148
61	Greedy Superstring	152
62	Shortest Common Superstring of Short Words	155
63	Counting Factors by Length	157
64	Counting Factors Covering a Position	160
65	Longest Common-Parity Factors	161
66	Word Square-Freeness with DBF	162
67	Generic Words of Factor Equations	164
68	Searching an Infinite Word	166
69	Perfect Words	169
70	Dense Binary Words	173
71	Factor Oracle	175
5	Regularities in Words	180
72	Three Square Prefixes	181
73	Tight Bounds on Occurrences of Powers	183
74	Computing Runs on General Alphabets	185
75	Testing Overlaps in a Binary Word	188
76	Overlap-Free Game	190
77	Anchored Squares	192
78	Almost Square-Free Words	195
79	Binary Words with Few Squares	197
80	Building Long Square-Free Words	199
81	Testing Morphism Square-Freeness	201
82	Number of Square Factors in Labelled Trees	203
83	Counting Squares in Combs in Linear Time	206
84	Cubic Runs	208
85	Short Square and Local Period	210
86	The Number of Runs	212
87	Computing Runs on Sorted Alphabet	214
88	Periodicity and Factor Complexity	219
89	Periodicity of Morphic Words	220
90	Simple Anti-powers	222
91	Palindromic Concatenation of Palindromes	224
92	Palindrome Trees	225
93	Unavoidable Patterns	227

6	Text Compression	230
94	BW Transform of Thue–Morse Words	231
95	BW Transform of Balanced Words	233
96	In-place BW Transform	237
97	Lempel–Ziv Factorisation	239
98	Lempel–Ziv–Welch Decoding	242
99	Cost of a Huffman Code	244
100	Length-Limited Huffman Coding	248
101	Online Huffman Coding	253
102	Run-Length Encoding	256
103	A Compact Factor Automaton	261
104	Compressed Matching in a Fibonacci Word	264
105	Prediction by Partial Matching	266
106	Compressing Suffix Arrays	269
107	Compression Ratio of Greedy Superstrings	271
7	Miscellaneous	275
108	Binary Pascal Words	276
109	Self-Reproducing Words	278
110	Weights of Factors	280
111	Letter-Occurrence Differences	282
112	Factoring with Border-Free Prefixes	283
113	Primitivity Test for Unary Extensions	286
114	Partially Commutative Alphabets	288
115	Greatest Fixed-Density Necklace	290
116	Period-Equivalent Binary Words	292
117	Online Generation of de Bruijn Words	295
118	Recursive Generation of de Bruijn Words	298
119	Word Equations with Given Lengths of Variables	300
120	Diverse Factors over a Three-Letter Alphabet	302
121	Longest Increasing Subsequence	304
122	Unavoidable Sets via Lyndon Words	306
123	Synchronising Words	309
124	Safe-Opening Words	311
125	Superwords of Shortened Permutations	314
	<i>Bibliography</i>	318
	<i>Index</i>	332

Preface

This book is about algorithms on texts, also called algorithmic stringology. Text (word, string, sequence) is one of the main unstructured data types and the subject is of vital importance in computer science.

The subject is versatile because it is a basic requirement in many sciences, especially in computer science and engineering. The treatment of unstructured data is a very lively area and demands efficient methods owing both to their presence in highly repetitive instructions of operating systems and to the vast amount of data that needs to be analysed on digital networks and equipments. The latter is clear for information technology companies that manage massive data in their data centres but also holds for most scientific areas beyond Computer science.

The book presents a collection of the most interesting representative problems in stringology. They are introduced in a short and pleasant way and open doors to more advanced topics. They were extracted from hundreds of serious scientific publications, some of which are more than a hundred years old and some are very fresh and up to date. Most of the problems are related to applications while others are more abstract. The core part of most of them is an ingenious short algorithmic solution except for a few introductory combinatorial problems.

This is not just yet another monograph on the subject but a series of problems (puzzles and exercises). It is a complement to books dedicated to the subject in which topics are introduced in a more academic and comprehensive way. Nevertheless, most concepts in the field are included in the book, which fills a missing gap and is very expected and needed, especially for students and teachers, as the first problem-solving textbook of the domain.

The book is organised into seven chapters:

- ‘The Very Basics of Stringology’ is a preliminary chapter introducing the terminology, basic concepts and tools for the next chapters and that reflects six main streams in the area.
- ‘Combinatorial Puzzles’ is about combinatorics on words, an important topic because many algorithms are based on combinatorial properties of their input.
- ‘Pattern Matching’ deals with the most classical subject, text searching and string matching.
- ‘Efficient Data Structures’ is about data structures for text indexing. They are used as fundamental tools in a large number of algorithms, such as special arrays and trees associated with texts.
- ‘Regularities in Words’ concerns regularities that occur in texts, in particular repetitions and symmetries, that have a strong influence on the efficiency of algorithms.
- ‘Text Compression’ is devoted to several methods of the practically important area of conservative text compression.
- ‘Miscellaneous’ contains various problems that do not fit in earlier chapters but certainly deserve presentation.

Problems listed in the book have been accumulated and developed over several years of teaching on string algorithms in our own different institutions in France, Poland, UK and USA. They have been taught mostly to master’s students and are given with solutions as well as with references for further readings. The content also profits from the experience authors gained in writing previous textbooks.

Anyone teaching graduate courses on data structures and algorithms can select whatever they like from our book for their students. However, the overall book is not elementary and is intended as a reference for researchers, PhD and master’s students, as well as for academics teaching courses on algorithms even if they are not directly related to text algorithms. It should be viewed as a companion to standard textbooks on the domain. The self-contained presentation of problems provides a rapid access to their understanding and to their solutions without requiring a deep background on the subject.

The book is useful for specialised courses on text algorithms, as well as for more general courses on algorithms and data structures. It introduces all required concepts and notions to solve problems but some prerequisites in bachelor- or sophomore-level academic courses on algorithms, data structures and discrete mathematics certainly help in grasping the material more easily.