

| | |
|-------------------------|---|
| 1. Record Nr. | UNINA9910784735803321 |
| Autore | Crochemore Maxime <1947-> |
| Titolo | Jewels of stringology [[electronic resource] /] / Maxime Crochemore, Wojciech Rytter |
| Pubbl/distr/stampa | Singapore ; ; River Edge, NJ, : World Scientific, 2002 |
| ISBN | 981-277-822-5 |
| Descrizione fisica | x, 310 p. : ill |
| Altri autori (Persone) | RytterWojciech |
| Disciplina | 005.1 |
| Soggetti | Computer algorithms Matching theory |
| Lingua di pubblicazione | Inglese |
| Formato | Materiale a stampa |
| Livello bibliografico | Monografia |
| Note generali | Bibliographic Level Mode of Issuance: Monograph |
| Nota di bibliografia | Includes bibliography references and index. |
| Nota di contenuto | 1. Stringology. 1.1. Text file facilities. 1.2. Dictionaries. 1.3. Data compression. 1.4. Applications of text algorithms in genetics. 1.5. Efficiency of algorithms. 1.6. Some notation and formal definitions. 1.7. Some simple combinatorics of strings. 1.8. Some other interesting strings. 1.9. Cyclic shifts and primitive words -- 2. Basic string searching algorithms. 2.1. Knuth-Morris-Pratt algorithm. 2.2. Boyer-Moore algorithm and its variations -- 3. Preprocessing for basic searchings. 3.1. Preprocessing patterns for MP and KMP algorithms. 3.2. Table of prefixes. 3.3. Preprocessing for Boyer-Moore algorithm. 3.4. * Analysis of Boyer-Moore algorithm -- 4. On-line construction of suffix trees. 4.1. Tries and their compact versions. 4.2. Prelude to Ukkonen algorithm. 4.3. Ukkonen algorithm -- 5. More on suffix trees. 5.1. Several applications of suffix trees. 5.2. McCreight algorithm -- 6. Subword graphs. 6.1. Directed acyclic graph. 6.2. On-line construction of subword graphs. 6.3. The reverse perspective. 6.4. Compact subword graphs -- 7. Text algorithms related to sorting. 7.1. The naming technique: KMR algorithm. 7.2. Two-dimensional KMR algorithm. 7.3. Suffix arrays. 7.4. Constructing suffix trees by sorting. 7.5. The Lowest-Common-Anccestor dictionary. 7.6. Suffix-Merge-Sort -- 8. Symmetries and repetitions in texts. 8.1. Searching for symmetric words. 8.2. Compositions of symmetric words. 8.3. Searching for square factors -- 9. Constant-space searchings. 9.1. Constant-space matching for easy patterns. 9.2. MaxSuffix-matching. 9.3. Computation |

of maximal suffixes. 9.4. Matching patterns with short maximal suffixes. 9.5. Two-way matching and magic decomposition. 9.6. Sequential sampling for unordered alphabets. 9.7. Galil-Seiferas algorithm. 9.8. Cyclic equality of words -- 10. Text compression techniques. 10.1. Substitutions. 10.2. Static Huffman coding. 10.3. Dynamic Huffman coding. 10.4. Factor encoding -- 11. Automata-theoretic approach. 11.1. Aho-Corasick automaton. 11.2. Determinizing automata. 11.3. Two-way pushdown automata -- 12. Approximate pattern matching. 12.1. Edit distance. 12.2. Longest common subsequence problem. 12.3. String matching with errors. 12.4. String matching with don't care symbols -- 13. Matching by dueling and sampling. 13.1. String matching by duels. 13.2. String matching by sampling -- 14. Two-dimensional pattern matching. 14.1. Multi-pattern approach. 14.2. Don't cares and non-rectangular patterns. 14.3. 2D-Pattern matching with mismatches. 14.4. Multi-pattern matching. 14.5. Matching by sampling. 14.6. An algorithm fast on the average -- 15. Two-dimensional periodicities. 15.1. Amir-Benson-Farach algorithm. 15.2. Geometry of two-dimensional periodicities. 15.3. * Patterns with large monochromatic centers. 15.4. * A version of the Galil-Park algorithm -- 16. Parallel text algorithms. 16.1. The abstract model of parallel computing. 16.2. Parallel string-matching algorithms. 16.3. * Splitting technique. 16.4. Parallel KMR algorithm and application. 16.5. Parallel Huffman coding. 16.6. Edit distance - efficient parallel computation -- 17. Miscellaneous. 17.1. Karp-Rabin string matching by hashing. 17.2. Shortest common superstrings. 17.3. Unique-decipherability problem. 17.4. Parameterized pattern matching. 17.5. Breaking paragraphs into lines.

Sommario/riassunto

The term "stringology" is a popular nickname for text algorithms, or algorithms on strings. This book deals with the most basic algorithms in the area. Most of them can be viewed as "algorithmic jewels" and deserve reader-friendly presentation. One of the main aims of the book is to present several of the most celebrated algorithms in a simple way by omitting obscuring details and separating algorithmic structure from combinatorial theoretical background. The book reflects the relationships between applications of text-algorithmic techniques and the classification of algorithms according to the measures of complexity considered. The text can be viewed as a parade of algorithms in which the main purpose is to discuss the foundations of the algorithms and their interconnections. One can partition the algorithmic problems discussed into practical and theoretical problems. Certainly, string matching and data compression are in the former class, while most problems related to symmetries and repetitions in texts are in the latter. However, all the problems are interesting from an algorithmic point of view and enable the reader to appreciate the importance of combinatorics on words as a tool in the design of efficient text algorithms. In most textbooks on algorithms and data structures, the presentation of efficient algorithms on words is quite short as compared to issues in graph theory, sorting, searching, and some other areas. At the same time, there are many presentations of interesting algorithms on words accessible only in journals and in a form directed mainly at specialists. This book fills the gap in the book literature on algorithms on words, and brings together the many results presently dispersed in the masses of journal articles. The presentation is reader-friendly; many examples and about two hundred figures illustrate nicely the behaviour of otherwise very complex algorithms.
