

The Book Review Column¹

by **Nicholas Tran** (ntran@scu.edu)

Department of Mathematics & Computer Science, Santa Clara University



1 Notable New Releases

Computational Intractability: A Guide to Algorithmic Lower Bounds (The MIT Press, forthcoming), by Erik Demaine (MIT), William Gasarch (University of Maryland), and Mohammad Hajiaghayi (University of Maryland), is a sequel to Garey & Johnson's classic book *Computers and Intractability: A Guide to the Theory of NP-Completeness*. A draft is available online at <https://hardness.mit.edu>.

The Delicate Art of Brute Force: How to Compute It When You Can't Solve It (Princeton University Press, 2026), by Paul Nahin, looks at a series of apparently intractable real-world problems and shows the thought process that allows computational solution.

Exploring Computational Geometry: Theory and Python Implementations (Springer, 2026), by Adis Alihodžić (University of Sarajevo), presents a modern approach to computational geometry through the lens of Python programming.

2 This Column

James Rauff shares stories about mathematical equations told by Robin Wilson in his book *Sum Stories: Equations and Their Origins* (Oxford University Press, 2025).

I add *An Introduction to String Algorithms* (Princeton University Press, 2026) by Carl Kingsford to my collection of essential algorithms textbooks.

3 How to Contribute

Read a good book over the summer? Why not write a review for *SIGACT News*? Either choose from the books listed below or propose your own. In either case, the publisher will send you a free copy of the book. Guidelines and a LaTeX template can be found at <https://algoplexity.com/~ntran>.

BOOKS THAT NEED REVIEWERS FOR THE SIGACT NEWS COLUMN

Algorithms & Complexity

¹©2026 Nicholas Tran

1. Brody, J. (2025). *The Joy of Quantum Computing: A Concise Introduction*. Princeton University Press.
2. Dalzell, A., McArdle, S., Berta, M., Bienias, P., Chen, C.-F., Gilyén, A., Hann, C., Kaskoryano, M., Khabiboulline, E., Kubica, A., Salton, G., Wang, S., & Brandão, F. (2025). *Quantum Algorithms: A Survey of Applications and End-to-end Complexities*. Cambridge University Press.
3. Erciyes, K. (2025). *Guide to Distributed Algorithms: Design, Analysis and Implementation Using Python*. Springer.
4. Morazán, M. T. (2025). *Programming-based Formal Languages and Automata Theory: Design, Implement, Validate, and Prove*. Springer.
5. Herbert, S. (2026). *Quantum Computing: Foundations and Practice*. Oxford University Press.
6. Demaine, E., Gasarch, W., & Hajiaghayi, M. (forthcoming). *Computational Intractability: A Guide to Algorithmic Lower Bounds*. The MIT Press.

Computability & Logic

1. Badia, G., Crossley, J. N., & Stillwell, J. C. (2026). *What is Mathematical Logic?, 2nd ed.* Oxford University Press.

Miscellaneous Computer Science & Mathematics

1. O'Rourke, J. (2025). *The Mathematics of Origami*. Cambridge University Press.
2. Meister, M., Lee, K. H., & Portugues, R. (2025). *Mathematical Biology*. The MIT Press.
3. Nahin, P. (2026). *The Delicate Art of Brute Force: How to Compute It When You Can't Solve It*. Princeton University Press.

Data Science

1. Alpaydm, E. (2025). *Fundamentals of Probability and Statistics for Machine Learning*. The MIT Press.

Discrete Mathematics and Computing

1. Devadoss, S., & O'Rourke, J. (2025). *Discrete and Computational Geometry, 2nd ed.* Princeton University Press.
2. Athanasiadis, C. A. (2026). *Discrete Mathematics: A Combinatorial Approach*. Springer.
3. Alihodžić, A. (2026). *Exploring Computational Geometry: Theory and Python Implementations*. Springer.

Cryptography and Security

1. Garfinkel, S. (2025). *Differential Privacy*. The MIT Press.
2. Beyne, T., & Rijmen, V. (2025). *Linear Cryptanalysis*. Cambridge University Press.
3. Rosulek, M. (2026). *The Joy of Cryptography: An Undergraduate Course in Provable Security*. MIT Press.

Combinatorics and Graph Theory

1. Fijalkow, N. (Ed.) (2026). *Games on Graphs: From Logic and Automata to Algorithms*. Cambridge University Press.
2. Bollobás, B., & Morris, R. (2026). *Basic Graph Theory*. Cambridge University Press.



Review of ¹

Sum Stories: Equations and Their Origins

Robin Wilson

Oxford University Press, 2025

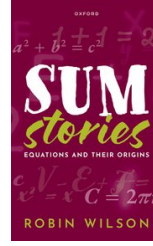
357 pages, \$29.64 Hardcover, \$19.24 eBook

Review by

James V. Rauff (jrauff@millikin.edu)

School of Mathematics and Computational Sciences

Millikin University, Decatur, IL



1 Overview

Sum Stories: Equations and Their Origins is a book about the history of selected mathematical topics written for a general audience. The author, Robin Wilson, is a well-known, widely published and respected graph theorist and historian of mathematics. He is an Emeritus Professor of Pure Mathematics at the Open University and a former President of the British Society for the History of Mathematics. Each chapter of *Sum Stories* bears an equation as its title. That equation serves as a launching point for a presentation of related mathematical results. Wilson prefers to call his chapters “stories” because they give historical context to the equation. I’ll do the same in this review. Taken as a whole, the stories in the book serve as a summary of most of the topics of undergraduate mathematics, with a few that are usually only seen by postgraduates. The major mathematical personalities behind the development of the topics and short descriptions of the mathematical motivation for these developments are included in each story. Proofs and other technical results are separated into boxes that can easily be skipped. The separate stories are not presented in strict chronological order. Each story unfolds from its earliest attestation of the equation to contemporary developments or applications. There is a rough grouping of the stories into four broad categories: school algebra, numbers, and geometry; discrete mathematics; (pre)calculus; and advanced mathematics.

2 Summary of Contents and Chapter Highlights

Stories 1-5: School Algebra, Numbers, and Geometry

Story 1 is $1 + 1 = 2$. Herein we find a discussion of early counting and numeration systems, including those of Egypt, Mesopotamia, Mesoamerica, China, India, Greece, and Rome. The integers, rationals, irrationals, and real numbers are then introduced. Included is the standard proof of the irrationality of $\sqrt{2}$ and Dedekind’s definition of a real number. It is clear from the

¹©2026 James Rauff

outset that Wilson will not shy away from more advanced topics that he deems relevant to his story.

Story 2 bears the title $\alpha + \beta + \gamma = 180^\circ$ and presents highlights from Greek geometry and tiling of the plane. Non-Euclidean geometry is also introduced in this chapter.

Story 3 concerns the familiar Pythagorean Theorem: $a^2 + b^2 = c^2$. Several historical proofs of the theorem (including U.S. President Garfield's) are included as well as extensions of the theorem into trigonometry. Wilson then discusses Pythagorean triples and how primitive triples can be generated. (Oddly, he omits a discussion of Plimpton 322, the Babylonian tablet from c. 1800 BCE that tabulates Pythagorean triples long before Pythagoras.) This chapter concludes with Fermat's Last Theorem.

Story 4 is $C = 2\pi r$ and $A = \pi r^2$. As one might expect, this story is all about the nature of π and the approximations made for it over time, from the polygonal approximations of Archimedes to series calculations to computer approximations. Also included are Buffon's needle experiment and the infamous Indiana π bill.

Story 5 is $ax^2 + bx + c = 0$ and is about the search for solutions of polynomial equations. The usual characters appear in this chapter: the Rhind papyrus, al-Khwārizmī, Omar Khayyām, Cardano, Tartaglia, Abel, and Galois.

Stories 6-10: Discrete Mathematics

The equation for **Story 6** is $\phi^2 = \phi + 1$, and the topic is the golden ratio. Wilson explores the appearance of the constant $\frac{1+\sqrt{5}}{2}$ in geometry, art, music, architecture, Penrose tilings, and the Fibonacci sequence.

Story 7 ($x^2 = x$) is about Boolean algebra, set theory, and Shannon's switching circuits.

Story 8 takes a deep dive into Mersenne primes, perfect numbers, and Fermat primes. Here Wilson gives the classic proof of the infinitude of primes as well as the stories of the search for primes, including GIMPS (the Great Internet Mersenne Prime Search). The equations $M_n = 2^n - 1$ and $F_n = 2^{2^n} + 1$ make up the title of this story.

Combinatorics is the topic of **Story 9**. Combinations and permutations, Pascal's arithmetical triangle, the binomial theorem, and error-correcting codes are some of the features of this story whose title is $C(n, r) = C(n-1, r) + C(n-1, r-1)$.

Euler's polyhedral formula, $V - E + F = 2$, is the name of **Story 10**. As might be expected, the Platonic and Archimedean solids are featured here along with Kepler's vision of the Solar System. What may be unexpected are the appearances of Kepler-Poinsot polyhedra and fullerenes.

Stories 11-15: Precalculus and Calculus

After the venture into discrete mathematics, Wilson's stories return to the familiar (and not so familiar) topics of precalculus and calculus.

Story 11 ($y = mx + c$) looks at the Cartesian coordinate system and the graphs of conics and other curves, including the conchoid of Nicomedes and the witch of Agnesi. Polar coordinates are also defined in this story.

Story 12 is about the invention of calculus and the relationship between tangents and areas as expressed in the story's title, $\int f'(x) dx = f(x)$, the Fundamental Theorem of Calculus.

Story 13 explores logarithms and exponentials and their properties, the constant e , and exponential growth. The title $\log(a \times b) = \log(a) + \log(b)$ focuses our attention on the power of logarithms to change a problem of multiplication into one of addition.

Story 14 ($i^2 = -1$) brings the imaginary and complex numbers into the mix along with the Fundamental Theorem of Algebra and Hamilton's quaternions.

The unexpected (to undergraduates) connection between the complex numbers and trigonometry expressed in Euler's celebrated equation $e^{ix} = \cos x + i \sin x$ and its very special case $e^{i\pi} + 1 = 0$ is the focus of **Story 15**.

Stories 16-18: Advanced Mathematics

The three stories that conclude the book are the most mathematically challenging to the reader.

Story 16 ($1 + 1/4 + 1/9 + 1/16 + 1/25 + \dots =$) is about the Basel Problem, the zeta function, and the Riemann hypothesis.

Story 17 ($z_{n+1} = z_n^2 + c$) plunges into the wonders of fractal geometry and the creatures that inhabit that world: the Koch snowflake, Sierpinski's triangle, Hilbert's space-filling curve, Julia and Mandelbrot sets, and fractional dimensions.

Story 18 concerns infinity, countable and uncountable sets, the hierarchy of the alephs, and the continuum hypothesis. The very appropriate equation that inspires this story is $\aleph_0 + \aleph_0 = \aleph_0$.

The book concludes with a list of suggested **Further Reading**. These include several standard history of mathematics texts as well as popular, technical, and original source books devoted to the equations of the stories.

3 My Opinion

Robin Wilson has an impressive knowledge of a broad range of mathematical topics and their histories. *Sum Stories* is densely packed. Each "story" contains personalities, references to historical texts, images, and mathematical derivations. (The title *Sum Stories* may be viewed as a pun on "some stories," and that would be more accurate since not all of the title equations involve sums.) The book can serve as a first go-to reference for the history of pretty much every topic in the undergraduate mathematics canon.

Although the chapters of the book are called "stories," they are more like bullet-pointed outlines. We read about an abundance of things, but the "stories" lack the flow one would expect of a narrative. For example, **Story 4** is not about its eponymous equations $C = 2\pi r$ and $A = \pi r^2$. Wilson is forthright about this and tells us that the story is about π . The "story" covers where the constant π arises, cross-cultural approximations of π , infinite series that produce π , computer approximations of π , and π miscellany. All of this is good stuff, but, in my opinion, it really isn't a story. Perhaps "snapshot" would be a better description. We have a collection of verbal and mathematical images organized around a theme. The equation that serves as the story's title may be the theme, but often that theme is a field of mathematics or a mathematical constant.

Sum Stories is not a book that one would read cover to cover. It is one that a reader could use as a starting point for understanding the development of a particular topic of mathematics and for why that topic is relevant to today's mathematics. It could serve as a supplementary text for a history of mathematics course. It would also be a handy reference to have on one's shelf to hand

to a student who asks a question like, “What is the golden ratio, where did it come from, and why should I care about it?”

An Introduction to String Algorithms

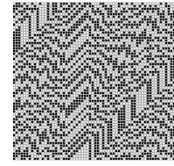
Carl Kingsford

Princeton University Press, 2026
376 pages, \$70 Hardcover, \$49 eBook

Review by

Nicholas Tran

Department of Mathematics and Computer Science
Santa Clara University



An Introduction to
String Algorithms
Carl Kingsford



1 Overview

As noted in the Preface, this book grew out of a semester course on string algorithms taught by the author at Carnegie Mellon University. Aimed at PhD students, the course provides a comprehensive high-level overview of string algorithms, with a slight emphasis on bioinformatics applications.

The book consists of 35 chapters and is organized into six parts: exact string matching, edit distance, data structures, sketching, graphs and generative models, and miscellaneous topics. A vast range of topics is covered, from classic algorithms on string matching and string alignment to modern string data structures and generative models. Averaging about ten pages per chapter, the book features one or two main ideas per topic, providing to-the-point explanations in plain English, figures, and illustrative examples to the mathematically mature reader. Each chapter ends with a summary and notes on further reading and references plus a handful to a dozen exercises.

This book focuses mainly on efficient and deterministic algorithms. It doesn't cover parallel, online, or quantum string algorithms, nor does it address theoretical questions studied in string combinatorics.

2 Summary of Contents

Chapter 1 motivates the study of string algorithms, establishes basic string notations and definitions, and lays out a roadmap of the remainder of the book, which consists of six parts.

Part I: Exact Matching

This part covers classic linear-time algorithms for the string matching problem: given a string called the *text* and one or more strings called the *patterns*, find all occurrences of the patterns in the text. **Chapter 2** introduces the Z-values of a string, popularized by Dan Gusfield's textbook [2] as a fundamental quantity for many string algorithms. **Chapter 3** shows how to implement

¹©2026 Nicholas Tran

the Boyer-Moore algorithm using the Z-values. **Chapter 4** gives two versions of the Knuth-Morris-Pratt algorithm: the traditional one based on deterministic finite automata, and the one based on Z-values. **Chapter 5** covers two seminumerical string matching algorithms: the probabilistic Rabin-Karp algorithm and the Shift-And algorithm due to Baeza-Yates and Gonnet. **Chapter 6** provides the original solution to the multi-pattern string matching problem due to Aho and Corasick.

Part II: Edit Distance

This part focuses on algorithms to determine whether two strings are mostly the same. **Chapter 7** defines the *edit distance* between two strings as the minimum number of insertions, deletions, and substitutions of characters in one string to turn it into the other. The *string alignment* problem for two given strings is to insert gaps into the strings to make them equal in length so that the edit distance between the modified strings is minimized. The chapter describes the well-known quadratic-time dynamic programming solution to this problem and its local and semi-global variants.

Chapter 8 examines a linear-space solution to the string alignment problem due to Hirschberg. **Chapter 9** explains how to get an $O(n^2/\lg n)$ -time solution to the edit distance problem when the cost of a mismatch, insertion, or deletion is 1 using the so-called “Four Russians” trick. A unique feature is its mention of the surprising connection between the strong Exponential Time Hypothesis and the lower bound on solutions to the edit distance problem.

Chapter 10 treats the edit distance variant with general gap penalties, e.g., multiple adjacent gaps cost less than the sum of the individual costs. **Chapter 11** considers solutions to the edit distance problem whose running time is dependent on the output (these run faster than quadratic time when the distance is small).

Chapter 12 is about the NP-complete problem of *multiple sequence alignment*, which asks for the minimum cost of aligning multiple strings. The chapter describes the Star MSA approximation algorithm that produces an alignment whose cost is within a factor of 2 of the optimal alignment.

Part III: Data Structures

Chapter 13 defines *suffix trees* and shows how they can be used to efficiently solve many string problems, such as string matching, longest common substrings of two strings, string matching with multiple patterns, and all-pairs suffix-prefix matches for a set of strings. **Chapter 14** explores more advanced applications of suffix trees, including Lempel-Ziv compression, longest common extension, finding palindromes, finding all k -mismatch occurrences of a pattern, and longest common substring with online queries. **Chapter 15** explains Ukkonen’s algorithm to construct the suffix tree of a given string in linear time.

Chapter 16 defines *suffix arrays* and the related *longest common prefix arrays* and shows that, together, they can be converted to suffix trees and vice versa in linear time. It also demonstrates how to use suffix arrays to solve the string matching problem in linear time. **Chapter 17** gives two algorithms to construct suffix arrays, the simpler one running in time $O(n \lg n)$, and the other in linear time.

The second half of the book (Chapters 18-35) covers topics not commonly found in other string algorithm textbooks.

Chapter 18 introduces the Burrows-Wheeler Transform, explains how to compute and invert it in linear time, and how to use it to solve the string matching problem and to perform compression.

Chapter 19 begins coverage of the so-called succinct data structures with a discussion of the RRR bit vector, which represents a read-only bit vector $B[1..n]$ and supports two operations: $\text{rank}_{0/1}(i)$ returns the number of 0/1 bits up to position i in the vector, and $\text{select}_{0/1}(j)$ returns the position of the j^{th} 0/1 bit in the vector. The value of the i^{th} bit is given by $\text{rank}_1(i) - \text{rank}_1(i-1)$. This data structure requires $nH_0(B) + o(n)$ bits of storage, and each supported operation takes constant time.

Chapter 20 considers *wavelet trees*, which extend RRR bit vectors to sequences S over an arbitrary alphabet Σ . Here the space usage is $nH_0(S) + o(n \log |\Sigma|)$ and the running time of rank and select is $O(\log |\Sigma|)$.

Chapter 21 is a brief overview of the *FM index*, which is a searchable compressed version of the BWT of a given string, implemented using wavelet trees.

Chapter 22 is another unique feature of this book, offering a summary of three ways texts are stored by text editors to support fast insertion and deletion: *gap buffers*, *piece tables*, and *ropes*.

Part IV: Sketching

This part introduces three ways to solve some string problems using little space, provided we tolerate some errors with small probability. **Chapter 23** explores how the collision probabilities of certain hash function families can be used as a proxy for the similarity between two given strings. Examples are given for the Jaccard similarity, the edit distance, and cosine-related similarity.

Chapter 24 examines *Bloom filters*, a data structure that maintains a collection of items and supports two operations: $\text{add}(x)$ adds x to the collection, and $\text{query}(x)$ returns **true** if x is in the collection. This is done using a number of hash functions with the same range. An element x is considered to be in the collection if all of the functions hash x to a position containing 1. The chapter establishes the number of hash functions to use to minimize the false positive probability. Some extensions to Bloom filters are considered.

Chapter 25 presents the *minimizers* method to quickly sample multiple strings to discover common substrings of a given length between any pair. This method was originally introduced for plagiarism detection but has found applications in bioinformatics.

Part V: Graphs and Generative Models

This part covers probabilistic and grammar-based models that generate strings, together with graph-based representations of sets of strings. **Chapter 26** covers formal grammars that define the Chomsky hierarchy: regular, context-free, context-sensitive, and unrestricted. **Chapter 27** examines *hidden Markov models*, which are probabilistic generators of strings using finite state machines. The *Viterbi* algorithm and the Forward-Backward algorithm are explained. Methods such as the Baum-Welch algorithm for training a hidden Markov model on labeled data are also covered.

Chapter 28 addresses *stochastic context-free grammars*, which are probabilistic generators of strings using context-free grammars. The Inside-Outside algorithms (analogs of the Forward-Backward algorithms for hidden Markov models) and Cocke-Younger-Kasami algorithms are presented along with methods to train a stochastic context-free grammar on labeled data.

Chapter 29 introduces *genome graphs* as a way to encode a set of strings as paths in a labeled graph. There are several types (de Bruijn graphs, variation graphs, and alignment graphs), each designed for a task such as genome assembly or sequence alignment.

Chapter 30 is also about storing a set of strings as a path in a *string tree* whose BWT-like properties allow efficient searching for a substring that appears in one or more of the encoded strings.

Chapter 31 discusses the *transformer* neural network architecture based on the *attention* mechanism.

Part VI: Miscellaneous

Chapter 32 looks at *Huffman codes*. **Chapter 33** presents a 4-approximation algorithm for the NP-complete problem of finding a *shortest superstring* of a set of given strings. The approximation algorithm is based on the related NP-complete problem of Traveling Salesman.

Chapter 34 gives three NP-hardness proofs. It is shown that Hamiltonian Path reduces to Shortest Superstring, Vertex Cover reduces to Shortest Common Supersequence, and Shortest Common Supersequence reduces to Multiple Sequence Alignment.

Chapter 35 points out a few algorithm design strategies that have been useful in solving string problems.

3 My Opinion

This book admirably achieves its goal of serving as a modern on-ramp to advanced studies in string algorithms. Collected here in one half of the volume are almost all of the topics found in standard texts in string algorithms such as [2, 1]: exact and inexact matching, suffix data structures, and compression. I would have liked to see a chapter on string combinatorics that covers the Lyndon-Schützenberger and Fine-Wilf theorems, since these results are often used in proofs of string algorithms that the reader may encounter later on, but I understand that the author wanted to keep the book focused on efficient algorithms rather than theoretical properties of strings. The other half of the book is devoted to material rarely found in competing texts, such as RRR bit vectors, wavelet trees, the FM index, minimizers, genome graphs, string trees, and transformers. I especially enjoyed the lucid explanations of transformers and the attention mechanism, and the way the book situates them alongside the earlier generative sequence models—hidden Markov models and stochastic context-free grammars. Some readers may question whether neural architectures belong in a string algorithms text, but I found their inclusion well motivated.

The presentation of some of the material in the first half of this book follows that of [2]. Even so, I appreciate the author’s unique perspective, such as his explanations of the DFA-based KMP algorithm and the “Four Russians” trick. To explain complex ideas using ten pages per chapter on average, the author relies effectively on figures and examples and avoids excessive mathematical notation, making the material remarkably accessible.

My recommendation is that this book is a must-read for advanced undergraduates and graduate students in theoretical computer science, to be followed by [2] for more implementational details or [1] for a more theoretical treatment of string algorithms.

References

- [1] M. Crochemore, C. Hancart, and T. Lecroq. *Algorithms on Strings*. Cambridge University Press, 2007.

- [2] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences : Computer Science and Computational Biology*. Cambridge University Press, 1997.