The Book Review Column[1]

by Nicholas Tran
Department Mathematics & Computer Science
Santa Clara University
Santa Clara, CA 95053
ntran@scu.edu

I thank my predecessors Fred Green (2015-2022) and Bill Gasarch (1997-2015) for their kind support and assistance in getting me prepared for this new role. I hope to continue and build on their work to make these pages a home for book lovers of the TCS community.

1 This column

Bill starts off this column with his approbatory review of a triptych on *The Mathematics of Various Entertaining Subjects* edited by Jennifer Beineke and Jason Rosenhouse that include contributions from game masters Peter Winkler, Erik Demaine, and others.

Author Vašek Chvátal is no stranger to recreational mathematics, though his latest book *The Discrete Mathematical Charms of Paul Erdős* is all serious despite its Buñuelesque title. Peter Ross reviews this comprehensive survey of discrete mathematics of the famous itinerant problem solver leavened with many appealing anecdotes.

S. V. Nagaraj completes the theme of this column with his review of *Programming for the Puzzled* by Srini Devadas, which advocates learning to program, or perhaps to think algorithmically, through puzzle solving.

2 How to contribute

Please contact me to write a review! Either choose from the books listed on the next page, or propose your own. The latter is preferred and quicker, as I can ask the publisher to send it directly to you.

BOOKS THAT NEED REVIEWERS FOR THE SIGACT NEWS COLUMN

Algorithms

1. Knebl, H. (2020). *Algorithms and Data Structures: Foundations and Probabilistic Methods for Design and Analysis*. Springer.

2. Roughgarden, T. (2022). *Algorithms Illuminated: Omnibus Edition*. Cambridge University Press.

Miscellaneous Computer Science

1. Amaral Turkman, M., Paulino, C., & Müller, P. (2019). *Computational Bayesian Statistics: An Introduction* (Institute of Mathematical Statistics Textbooks). Cambridge University Press.

2. Nakajima, S., Watanabe, K., & Sugiyama, M. (2019). *Variational Bayesian Learning Theory*. Cambridge University Press.

3. Hidary, J. D. (2021). *Quantum Computing: An Applied Approach* (2nd ed.). Springer.

4. Apt, K. R., & Hoare, T. (Eds.). (2022). *Edsger Wybe Dijkstra: His Life, Work, and Legacy* (ACM Books). Morgan & Claypool.

5. Burton, E., Goldsmith, J., Mattei, N., Siler, C., & Swiatek, S. (2023). *Computing and Technology Ethics: Engaging through Science Fiction*. The MIT Press.

Discrete Mathematics and Computing

1. O'Regan, G. (2020). *Mathematics in Computing: An Accessible Guide to Historical, Foundational and Application Contexts*. Springer Publishing.

2. Rosenberg, A. L., & Trystram, D. (2020). *Understand Mathematics, Understand Computing: Discrete Mathematics That All Computing Students Should Know*. Springer Publishing.

3. Liben-Nowell, D. (2022). *Connecting Discrete Mathematics and Computer Science* (2nd ed.). Cambridge University Press.

Cryptography and Security

1. Oorschot, P. V. C. (2020). *Computer Security and the Internet: Tools and Jewels* (Information Security and Cryptography). Springer.

Combinatorics and Graph Theory

1. Golumbic, M. C., & Sainte-Laguë, A. (2021). *The Zeroth Book of Graph Theory: An Annotated Translation of Les Réseaux (ou Graphes)—André Sainte-Laguë (1926)* (Lecture Notes in Mathematics). Springer.

2. Beineke, L., Golumbic, M., & Wilson, R. (Eds.). (2021). *Topics in Algorithmic Graph Theory* (Encyclopedia of Mathematics and its Applications). Cambridge University Press.

Programming etc.

1. Nielson, F., & Nielson, R. H. (2019). *Formal Methods: An Appetizer*. Springer.

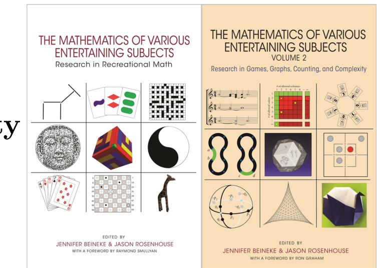
2. Sanders, P., Mehlhorn, K., Dietzfelbinger, M., & Dementiev, R. (2019). *Sequential and Parallel Algorithms and Data Structures: The Basic Toolbox*. Springer.

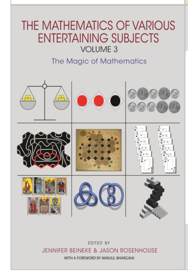
Miscellaneous Mathematics

1. Kurgalin, S., & Borzunov, S. (2022). *Algebra and Geometry with Python*. Springer.

Joint Review of[1]

The Mathematics of Various Entertaining Subjects:
Research in Recreational Math
and
The Mathematics of Various Entertaining Subjects:
Vol. 2. Research in Games, Graphs, Counting, and Complexity
and
The Mathematics of Various Entertaining Subjects
Vol. 3. The Magic of Mathematics





Edited by Jennifer Beineke and Jason Rosenhouse
Publisher: Princeton University Press
Numbers of Pages: 268, 386, 320
Year: 2016, 2017, 2019

Review by
William Gasarch (`gasarch@umd.edu`)
Department of Computer Science
University of Maryland, College Park

1 Introduction

Each of these three books is a collection of articles on recreational math. Or are they? What is recreational math? The following three points are neither necessary or sufficient for an article to be on recreational mathematics; however, it is a good guide.

1. The problem being discussed is accessible to a layperson. So the topic is often Combinatorics or Number Theory, but never Algebraic Geometry or Topos.

2. The answer to the problem can be understood by a layperson.

3. This one is optional: A layperson has a chance to solve the problem themselves.

4. I've used the term layperson without defining it. And I won't.

Ray Smullyan wrote the foreword to Volume 1, which is appropriate, since it is often said that

Ray Smullyan worked in both recreational and serious mathematics and made a mockery of the distinction between the two.

In this review I will describe a few articles from each book and discuss how recreational they are. In light of the quote above, I will not try too hard to distinguish "recreational math" from "serious math."

Why study recreational math?

1. Sometimes it leads to serious math, as this book proves.

2. Sometimes it serves as the starting point for more serious math.
3. Sometimes it can be used to get youngsters interested in math. Many people of my generation (I claim I am 62, though at one time Wikipedia thought I was 10 years older than I am) read Martin Gardner's recreational mathematics column in *Scientific American* and were inspired by it.
4. BECAUSE IT IS FUN!

2 Vol 1: Research in Recreational Mathematics

2.1 "Heartless Poker" by Dominic Lanphier and Laura Taalman

No, heartless poker is not poker played without mercy (I am not sure what that means). It refers to playing poker with a deck that has no hearts. This chapter is not really about that. It is about what happens if you vary (a) the number of ranks, (b) the number of suits, (c) the size of a hand.

1. In standard poker (13 ranks, 4 suits, 5-card hand) a straight (5 cards in increasing rank order, no wrap-around) is more likely than a flush (all of the same suit), which is more likely than a full house (a pair and 3-of-a-kind). But the probabilities are close together (straight ~ 0.00392, flush ~ 0.00196, full house ~ 0.00144.) This is recreational.
2. By changing the number of ranks and the number of suits, can you get all six possible orderings of what is more likely? You can! This is perhaps recreational.
3. Can you get all three probabilities to be the same ? No. This is perhaps recreational.
4. Can you get any two of the probabilities to be the same ? No. This requires using some hard number theory as a black box, so it is perhaps not recreational.

2.2 "Analysis of Crossword Puzzle Difficulty Using a Random Graph Process" by John K. McSweeney

Given a crossword puzzle, how hard is it to solve? This article *is not* about complexity theory, $n \times n$ grids, or NP-completeness. This article *is* about actual crossword puzzles from the newspaper that you do over breakfast. Or if it is hard then over breakfast, lunch, and dinner.

There are two factors that make a crossword puzzle hard.

1. The difficulty of the clues. And if a word intersects with several others, then the difficulty of its clue given partial information. The article models this by first giving, for each clue, a threshold that tells you what fraction of the letters needs to be there in order to solve it (recreational). Later, probability distributions are used for thresholds (perhaps not recreational).
2. The structure. This is modeled with a bipartite graph where the ACROSS words are on the left, the DOWN words are on the right, and if they intersect there is an edge between them.

The article looks at which properties of the bipartite graph make a puzzle easier or harder and applies its findings on *real examples*.

This article is easy to read if you skim some of the random process stuff. Nevertheless, reading it is easier than doing the Sunday *New York Times* crossword puzzle.

2.3 "The Cookie Monster Problem" by Leigh Marie Braswell and Tanya Khovanova

There are k jars of cookies. Jar 1 has c_1 cookies, ..., jar k has c_k cookies. A cookie monster wants to eat all of the cookies. He will do this in a sequence of moves. On move j he will pick a number a_j and a set of jars $I \subseteq \{1, \ldots, k\}$ and eat a_j cookies from each jar in I (the jars in I must all have at least a_j cookies).

Here is an example: The jars have (1, 5, 9, 10) cookies. On move 1 the monster eats 9 cookies from the 3rd and 4th jar, so now the jars are (1, 5, 0, 1). On move 2 he eats 1 cookie from the 1st, 2nd, and 4th jar, so now the jars are (0, 4, 0, 0). On move 3 he eats 4 cookies from the 2nd jar, so now the jars are (0, 0, 0, 0). That took 3 moves. Can he do better? I leave that as an exercise.

More generally, given a set of jars, what is the *minimum* number of moves needed to eat all of the cookies?

The article gives many strategies and sequences of jars and proves several theorems. They are all pleasant and easy. Definitely recreational. The Fibonacci numbers make a surprising appearance. Actually it's not surprising, since this section of the book is titled "Fibonacci Numbers." That aside, why are the Fibonacci numbers in this chapter? Because they provide an example of a sequence of jars that take a large number of moves.

3 Vol 2: Research in Games, Graphs, Counting, and Complexity

3.1 "The Cycle Prisoners" by Peter Winkler

There are n people in prison. They do not know n. They do have a leader L. The warden will do the following: Every night the prisoners write down a bit on a piece of paper. The warden collects them, looks at them, and then redistributes them to the prisoners via a cyclic shift. This goes on for a while until the prisoners all announce that they all know the n. At that point, they all go free! A few details: (a) the leader can broadcast instructions to the rest before the game begins, and (b) no communication between the prisoners once the game starts.

There are two startling facts about this game: (1) there is a solution, and (2) it has applications to distributed computing. I do not know which one is more startling.

The game is easy to describe (I just did). The solution is very very clever but understandable. Recreational! Though I will point out I doubt a layperson, or even a seasoned professional, can come up with the answer.

3.2 "Duels, Truels, Gruels, and the Survival of the Unfittest" by Dominic Lanphier

A *Duel* is as follows: (a) there are two players Alice and Bob who have guns, (b) Alice has probability p_A of hitting Bob, Bob has probability p_B of hitting Alice, (c) they shoot in order Alice, Bob, Alice, Bob, ... until there is only one person alive, (d) one shot that hits will kill.

The questions one asks are: (1) What is the probability that Alice survives? (2) What is the probability that Bob survives? (3) What is the expected number of rounds?

Duels has been well studied and are completely understood. What makes it easy is that (in contrast to Truels and n-ruels) neither player has a decision to make: Alice will aim at Bob, and Bob will aim at Alice.

What if you have three players?

A *Truel* is as follows: (a) there are three players Alice, Bob, Carol who have guns, (b) Alice has probability p_A of hitting whoever she aims at, Bob has probability p_B of hitting whoever he aims at, Carol has probability p_C of hitting whoever she aims at, (c) they shoot in order Alice, Bob, Carol, Alice, Bob, Carol, ... until there is only one person alive; whoever shoots can *choose* who they try to kill, (d) one shot that hits will kill.

Truels have been studied but are not fully understood. What makes it hard is that (in contrast to Duels) each player must *decide* who to aim at. An interesting aspect is that there are times the weakest shooter has the highest probability of surviving. For more on Truels see Brams & Kilgour [2] and Kilgour [3, 4, 5].

This article looks at three variants of these concepts.

1. n-uels: There are n people.

2. Gruels: There are n people and they are on the vertices of an undirected graph. Alice can aim to kill Bob only if there is an edge from Alice to Bob.

3. More than one shot is needed to kill a person.

This article covers a lot of old and new ground. The math is mostly easy: elementary algebra and probability. Some recurrences and some generating functions. Mostly recreational and perhaps a good introduction to recurrences and generating functions.

4 Vol 3: The Magic of Mathematics

4.1 "Probability in Your Head" by Peter Winkler

Eight problems in probability are presented which the author claims can be done in your head. All of the problems are (1) easy to understand and, (2) have solutions that are easy to understand. Indeed— I did not even need pencil and paper when reading the solutions. Are they problems that a layperson could do on their own? Hard to say.

Here is one of the problems (I won't give the answer):

Three points are chosen at random on a circle. What is the probability that there is a semicircle of that circle containing all three ?

In the Answer section the author gives the trick to being able to do it in your head and then states the (then easily proven) result for n points.

4.2 "Losing Checkers is Hard" by Jeffrey Bosboom, Spencer Congero, Erik D. Demaine, Martin L. Demaine, and Jason Lynch

This article talks about winning and losing checkers on a given $n \times n$ board that needs not be in the usual starting configuration. Robson [6] showed that determining who wins is EXPTIME-complete. The article explains what this means intuitively.

The article proves the following results:

1. Given a board and pieces on it, and it is Red's turn, can Red make a move so that Black is forced to win in one move? This problem is PSPACE-complete. You might want to lose at checkers if playing a child, a Wookiee, or a Wookiee child. See

https://www.starwars.com/video/let-the-wookiee-win

2. Consider the variant where if a player cannot capture a piece then they lose (this would not be interesting if the game has the standard initial placement of pieces). Determining if a given player can win is EXPTIME-complete.

3. Consider the variant where the players are cooperating and their goal is to eliminate all of the pieces of one color. The problem of determining if they can do this in 2 moves is NP-complete.

The problems are understandable to a layperson if they can accept on faith that NP-complete, PSPACE-complete, and EXPTIME-complete problems form a hierarchy of hard problems. The proofs would be hard for a layperson but understandable and interesting for the readers of this column.

4.3 "On Partitions into Squares of Distinct Integers Whose Reciprocals Sum to 1" by Max A. Alekseyev

Graham [1] proved the following:

> *For all $n \geq 78$ there exist $x_1 < \cdots < x_k \in \mathbb{N}$ such that (a) $\sum_{i=1}^{k} x_i = n$, and (b) $\sum_{i=1}^{k} \frac{1}{x_i} = 1$. Further, the lower bound 78 is tight.*

He conjectured that there was some N such that, for all $n \geq N$, his theorem with $\sum_{i=1}^{k} x_i$ replaced with $\sum_{i=1}^{k} x_i^2 = n$ holds.

This paper proves the conjecture. In particular they show the following:

> *For all $n \geq 8543$ there exist $x_1 < \cdots < x_k \in \mathbb{N}$ such that (a) $\sum_{i=1}^{k} x_i^2 = n$, and (b) $\sum_{i=1}^{k} \frac{1}{x_i} = 1$.*

This article proves both Graham's theorem and Graham's conjecture in a very motivated way. The term *induction* is never used so that a layperson can follow it.

The result is a nice combination of mathematics and computer science.

5 Opinion

Anyone reading this column will enjoy most of the articles in these books. They are a perfect storm of (a) problems of interest, (b) solutions of interest, (c) new to the reader, and (d) understandable to the reader.

But what about people who are not as mathematically mature as readers of this column? This question arises since the books advertise themselves as *Recreational Mathematics*, which implies that laypeople should also benefit. Looking over all of the articles with this in mind, we have an almost-perfect storm. They all hit a, b, and c. And they are all well written. Some of them fail on point d. Nevertheless, exposure is still good. For example, knowing that losing at checkers is a hard problem is still valuable, even if you don't quite know what that means, or the proof of this fact.

References

[1] R. Graham. A theorem on partitions. *Journal of the Australian Math Society*, 3, 1963. `http://www.math.ucsd.edu/~ronspubs/`.

[2] S. Grams and D. M. Kilgour. The Truel. *Mathematics Magazine*, 70:315–326, 1997.

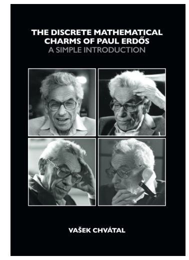
[3] D. M. Kilgour. The simultaneous Truel. *International Journal of Game Theory*, 1:229–242, 1972.

[4] D. M. Kilgour. The sequential Truel. *International Journal of Game Theory*, 4:151–172, 1974.

[5] D. M. Kilgour. Equilbrium points of infinite sequential Truels. *International Journal of Game Theory*, 6:167–180, 1977.

[6] J. M. Robson. N by N checkers is EXPTIME-complete. *SIAM Journal on Computing*, 13:252–267, 1984. `https://epubs.siam.org/doi/10.1137/0213018`.

Review of [1]



The Discrete Mathematical Charms of Paul Erdős
A Simple Introduction
by Vašek Chvátal

Cambridge University Press, 2021
Paperback, 266 pages, $29.99

Review by
Peter Ross (`pross@scu.edu`)

1 Introduction

Paul Erdős (1913-1996) was an essential singularity in the world of twentieth-century mathematicians. The author Vašek Chvátal, who met Erdős as a young undergraduate in the mid '60s, co-wrote three papers with him and maintained a special friendship for the rest of Erdős' life. Donald Knuth wrote, "Vašek Chvátal was born to write this one-of-a-kind book. Readers cannot help be captivated by the evident love with which every page has been written. The human side of mathematics is intertwined beautifully with first-rate exposition of first-rate results."

Chvátal's book is based on his lecture notes for a graduate course *Discrete Mathematics of Paul Erdős*, which he taught at Concordia University in Montreal. But the book goes way beyond notes for a graduate course, being a reference on the many topics it covers, with full proofs and recent results. The book's eleven chapters include ones on Bertrand's Postulate, Discrete Geometry, Ramsey's Theorem, van der Waerden's Theorem, Extremal Graph Theory, the Chromatic Number, Hamilton Cycles, and others. The Summary below includes representative theorems from three of these chapters.

Each chapter ends with some non-mathematical reminiscences from the author about Erdős and his life, thoughtfully set apart by a slightly shaded background. The Summary below also includes several lesser-known things about Erdős or quotes by him.

2 Summary

To give a flavor of the mathematics in the book, I've chosen a key theorem from each of three chapters that serves as the beginning point for deeper results in them.

<u>Bertrand's Postulate</u> (chapter 1)

In 1845 Joseph Bertrand conjectured that for every positive integer n, there is at least one prime number p such that $n < p \leq 2n$. Chebyshev proved this theorem (the term "postulate" is a misnomer) in 1852, and in 1931 the precocious Erdős gave an "elegant elementary proof," according to Chvátal, that was published as his first article. Erdős' proof was a tour de force of estimates for binomial coefficients, Legendre's formula for the exponents of primes in the unique factorization of integers, and a clever approach to showing non-constructively that there must be a prime between

n and $2n$. Erdős used a similar proof technique in much later papers unrelated to number theory, which became known as "the probabilistic method." The chapter includes sketches of other proofs of Bertrand's Postulate, including a somewhat-similar one by Ramanujan, and recent results and problems concerning primes.

Van der Waerden's Theorem (chapter 6)

In 1927 Bartel van der Waerden published a proof of a conjecture that he heard from the Dutch mathematician Han Baudet: For each positive integer k, if all positive integers are colored red or white there is an arithmetic progression of k terms with the same color. Forty-four years later van der Waerden published the article "How the proof of Baudet's conjecture was found," describing an afternoon working on it with Emil Artin and Otto Schreier at the University of Hamburg. They first proved some special cases, and then tried strong induction on a multicolor version of the conjecture. Chvátal's book uses a generalization of the conjecture involving two variables k and l, then presents a rather tedious proof of it by double induction on k and l. Motivated by van der Waerden's Theorem, Erdős and Turán proposed several conjectures that eventually led to Szemerédi's Theorem and his award of the Abel Prize in 2012. The chapter concludes with a discussion of a theorem of Hales and Jewett (1961) from Ramsey Theory, which implies van der Waerden's Theorem.

The Friendship Theorem (chapter 8)

In 1966 Erdős and collaborators A. Rényi and V.T. Sós proved that if in a (finite) graph every pair of vertices has precisely one common neighbor, then some vertex is adjacent to all of the other vertices. This theorem became known as the Friendship Theorem, as it asserts that if in a group of people every pair has exactly one friend in common, then someone in the group is a politician, that is, he's a friend of everyone. The proof is broken into two main cases, when the graph is regular and when it's not. A regular graph is one where every vertex has the same degree, that is, the same number of neighbors. The easier case to prove is when the graph is not regular. The harder case of a regular graph is broken into two subcases, in one of which a big gun is used, Baer's Theorem: every polarity in a finite projective plane maps some point to a line that contains this point. The book does give an alternate proof which avoids Baer's Theorem, but it's several pages long and tedious.

The remainder of the chapter involves strongly regular graphs and uses tools of linear algebra applied to the adjacency matrix of a graph, such as the eigenvalues of a square matrix and the Principal Axis Theorem for real symmetric square matrices.

3 Reminiscences about Erdős and his life

Part of the appeal of the book is its inclusion of many anecdotes about Erdős, some well-known and others discovered by the author in his friendship with Erdős for over three decades. For example, Erdős was an itinerant mathematician known for his quote, "Another roof, another proof," who also had a maxim that "property is a nuisance." Chvátal notes that Erdős was the antithesis of a snob, and an anarchist in the noblest sense of the term. According to a declassified FBI file on Erdős that's excerpted in an appendix, he did not apply for U.S. citizenship as, "I am stateless by political conviction." Erdős' naiveté was illustrated in the early years of World War II when he, S. Kakutani, and A. Stone were briefly arrested on Long Island for taking photos against a background of what turned out to be a secret radar station! The book includes a photo from the *New York Daily News* the next day with the headline "3 ALIENS NABBED AT SHORT-WAVE

STATION."

Not all quotes that are often attributed to Erdős are valid. In a footnote Chvátal says that the famous saying "A mathematician is a machine for turning coffee into theorems" was actually due to Erdős' friend A. Rényi. But the stories of Erdős' heavy use of Benzedrine and Ritalin to sustain 19-hour workdays and sharpen his concentration were true, particularly after his mother died in 1971. In 1979 mathematician Ron Graham, Erdős' good friend and part-time caretaker, bet him $500 that he couldn't stay off drugs for a month. Erdős won the bet but later told Graham that he had set mathematics back a month, and Erdős returned to using stimulants.

Chvátal asserts that the title of Paul Hoffman's 1998 biography *The Man Who Loved Only Numbers* was "a clear libel," as anyone who ever watched Erdős play ping-pong would attest. Three of the book's many photos, which are mostly taken from George Csicsery's film *N is a Number*, show Erdős playing table tennis.

A half-page in the book discusses how almost all mathematicians are the best in their schools but eventually discover they're no longer the smartest kids on the block. However Erdős, although a child prodigy, never suffered this shock and "remained the smartest kid on the block for the rest of his life." In spite of this he never became a snob, and Chvátal points out that Erdős provides an example that cooperation instead of rivalry makes mathematics rewarding and enjoyable, and his attitude made him lots of new friends and was infectious.

4 Opinion

The book does succeed in its main objective of surveying results of Erdős and others who laid the foundations of discrete mathematics. But its subtitle *A Simple Introduction* is only a half-truth as it's not simple. Even the relatively easy inclusion-exclusion principle for sets is only presented algebraically, with no hint of Venn diagrams for the simpler cases of 2 or 3 sets. The book would best serve as a reference book for computer science or mathematics professionals who want a comprehensive, up-to-date survey, and are willing to accept the book's many very detailed arguments. Interestingly, the only mention I spotted of what Erdős called THE BOOK (of perfect proofs of theorems that God maintains) was a reference in an appendix to Martin Aigner and Günter M. Ziegler's book *Proofs from THE BOOK* (Springer, 2014).

On the more positive side, many chapters begin with an enticing problem or theorem. For example, the chapter on Ramsey's Theorem opens with this problem from a Hungarian mathematics competition for high school students in 1947:

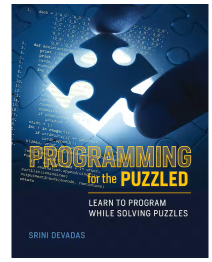
> Prove that in any group of six people, either there are three people who know one another or three people who do not know one another.

A version of this problem for six points in space appeared in the 1953 Putnam Mathematical Competition. Another nice feature for readers is that key definitions are often repeated in different chapters as REMINDERS, and they're also collected in Appendix B. The student-friendly twenty-three page Appendix A, "A Few Tricks of the Trade," presents mathematical tools, even with some proofs, involving powerful inequalities, factorials, binomial coefficients, the binomial and the hypergeometric distributions, and the like.

Finally, some nuts and bolts. There's an extensive Bibliography with 378 items, of which at least 55 are by Erdős, but the Index is a bit thin at only 5 pages. The book is attractive, with 4 photos of Erdős on its cover, and easy to lug around, being about 9.5" by 6.5" by 0.5".

Review of[1]

Programming for the Puzzled:
Learn to Program While Solving Puzzles
Srini Devadas
The MIT Press, 2017
272 pages, Paperback, $25.00



Review by
S. V. Nagaraj (svnagaraj@acm.org)
Vellore Institute of Technology
Chennai Campus, India

1 Introduction

This book is about learning to program while solving puzzles. The motivation for the author to use this approach is to gain better attention from students, many of whom do not like to program just for the sake of programming. As mentioned in the informative preface, this book reflects the author's "attempt at teaching programming by building a bridge between the recreational world of algorithmic puzzles and the pragmatic world of computer programming," because the same analytical skills required for puzzle solving are also needed for "translating specifications into programming constructs, as well as discovering errors in early versions of code, called the debugging process."

Twenty one often-familiar puzzles are discussed in the book. Each chapter begins with a puzzle's description followed by a solution given in the form of specification of the code that needs to be written. Programming constructs and algorithmic paradigms needed to realize the code are made comprehensible. It is assumed that the readers have only a high school level understanding of programming concepts. Programming exercises are provided at the end of each chapter. These exercises vary in difficulty and the amount of code that needs to be written. Some of the puzzles are contrived to make the readers learn some algorithmic idea, while others, called Puzzle Exercises, involve more effort, and some of them may be considered as advanced puzzles. The code for puzzle solutions is downloadable from the book's website [2]. Ancillary material and the code for all puzzle solutions are available to instructors. Supplementary materials for the first eleven puzzles are available on YouTube [3] and as open courseware [4].

2 Summary

The first chapter is seemingly about how to give instructions to groups of people so that they wear hats the right way. However, it is actually about run-length encoding and data compression.

The second chapter is about finding the right hour to attend a party so as to maximize the number of celebrities with whom we can interact.

The third chapter is about encoding and communicating information, but cloaked up in the form of a card trick.

The fourth chapter is about the eight queens problem, which is a classic computer science problem.

The fifth chapter is about breaking a crystal ball. You have to drop the crystal ball from a floor in a high rise and determine the maximum floor in it from which it will not break. A ball can't be reused once it has been broken.

The sixth chapter is about detecting a fake coin in a set of coins that look identical.

The seventh chapter is about finding square roots.

The eighth chapter is about guessing who is not going to turn up for dinner. The maximum number of guests should be invited but no two of them should dislike each other.

The ninth chapter is about selecting candidates with varied talents. This puzzle is actually an instance of the set-covering problem, which has a large number of applications.

The tenth chapter is about the n-queens problem.

The eleventh chapter is about tiling a courtyard with L-shaped tiles.

The twelfth chapter is about the towers of Brahma puzzle with a twist.

The next chapter is about a disorganized handyman who has a whole collection of nuts and bolts of different sizes in a bag. Each nut is unique and has a matching unique bolt, but the handyman mixed them up. The objective is to find the right pairs of nuts and bolts. In this chapter, the author mentions the fact that built-in functions in Python may appear to be faster not because they are algorithmically better but because they have been carefully written in a low-level language.

The fourteenth chapter is related to the Sudoku puzzle.

The fifteenth chapter is about counting the ways change may be given using known denominations.

The sixteenth chapter is titled: "Greed is good." The problem is to maximize the number of courses that a student can take in any given semester.

The seventeenth chapter is about grouping anagrams.

The eighteenth chapter is about memoization. The problem introduced here requires us to pick a subset of a given set of coins so as to maximize their sum, although we are not allowed to pick two adjacent coins.

The nineteenth chapter is titled: "A weekend to remember." Your friends have to be invited for dinner, which you host on consecutive nights. There are conditions though: Each of your friends must attend exactly one of the two dinners. In addition, if A dislikes B or B dislikes A, they cannot both be in the same dinner party. This puzzle is related to bipartite graphs.

The twentieth chapter is about six degrees of separation, an observation that "everyone is six or fewer steps away, by way of introduction, from any other person in the world." The breadth-first search algorithm is introduced here.

The last chapter is related to the twenty questions game albeit with a twist. The binary search tree data structure and a greedy algorithm are the key concepts of this chapter.

3 Opinion

The title of the book could confuse some of those hunting for books. A better title would perhaps be: *Learning algorithms through programming and puzzle solving.* Perhaps an even more apt title might be: *Python programming for the puzzled.* It should be stated that not all the puzzles may be of interest to all the readers. Solving all the puzzles will not be easy for many. The author has taken effort to foster creative thinking along with programming. The puzzles in the book may lead the readers to solve even more mind-blowing puzzles in the future.

One may ask whether there is a positive correlation between puzzle-solving performance and coding experience, and between the puzzle difficulty for humans and non-human AI solvers. An interesting article by Schuster et al. studies this [1]. They have created an interesting database of programming puzzles in Python [5] that contains a dataset of Python programming puzzles for teaching and evaluating an AI's programming proficiency. Schuster and others state that puzzles are often used to teach and evaluate human programmers. Many classic puzzles such as the Tower of Hanoi teach fundamental concepts such as recursion, and programming competition problems, sometimes referred to as puzzles, evaluate a participant's ability to apply these concepts. Puzzles are also often used to judge programmers in job interviews, and some puzzles such as the RSA-factoring challenge test the limits of state-of-the-art algorithms.

This book should definitely help its readers improve their puzzle-solving as well as programming and algorithm skills that apply in any language not just Python. I believe this book is surely much more interesting and easier to follow compared to other programming books. However, it should be mentioned that the book does not really describe the basic technical steps for writing Python programs. Those who have no programming experience should invariably opt for a more traditional programming book. The plus side of this book is that it focuses more on teaching how to think as a programmer rather than just teaching syntax.

Although some readers may feel that coding puzzles may be a suitable way ahead if they are aimed at the beginner, they may also feel that some advanced puzzles tend to rely on techniques that are perhaps obscure. This in the worst scenario could lead to bad programming practices. Solving puzzles may be good for two reasons: (i) there will be focus on reasonable length of code for learning a new programming language, and (ii) the reader may learn by doing. The author follows this idea. However, this approach may not work if a reader is completely new to a programming language such as Python. Puzzles by themselves often do not teach a reader the techniques needed to design good quality software. As a matter of fact, there are often many language-specific rules of thumb for constructing utilitarian and expandable applications. Often a reader is unlikely to get exposed to these by merely solving just puzzles. A professional coder may also wonder whether coding puzzles is indeed reflective of real-world production code. One may even go to the extent of saying that solving puzzles does not teach you how to write beneficial code, or perhaps maintainable code.

I believe this book would not be worth buying for readers who are unfamiliar with Python, not interested in puzzles, and don't like programming. The readers should be reasonably good programmers interested in problem solving using Python to get maximum benefit from this book. I believe that the objective of this book is more to get the learner to think like a programmer rather than to teach the learner how to program. Programmers are often concerned with mundane chores rather than puzzles. Some of the chapters do not highlight the important algorithmic paradigms they use. For example, the fifteenth chapter on offering change does not state in the beginning that greedy algorithms are used. Nevertheless, the book should be of interest to those who want to learn how to solve puzzles and simultaneously learn programming using Python.

References

[1] T. Schuster and A. Kalyan and A. Polozov and A. Kalai. Programming Puzzles. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 1, 2021.

[2] `https://mitpress.mit.edu/books/programming-puzzled`

[3] `https://www.youtube.com/playlist?list=PLUl4u3cNGP62QumaaZtCCjkID-NgqrleA`

[4] `https://ocw.mit.edu/`

[5] `https://github.com/microsoft/PythonProgrammingPuzzles`