

Book Reviews

MARK LAMOURINE AND MICHELE NELSON

Gnuplot in Action: Understanding Data with Graphs (Second Edition)

Phillipp K. Janert

Manning Publications Co., 2016, 372 pages

ISBN 978-1-63343-018-1

Reviewed by Mark Lamourine

Gnuplot has always been a chicken/egg tool for me. There are times when I have data series I'd like to plot, but I'm not fluent enough with Gnuplot to produce something useful quickly. Without interesting (read: urgent) data, I have more pressing things to do than to learn a new tool. When I saw that there was a new edition of *Gnuplot in Action* I thought it was time to try again.

Gnuplot is unrelated to the GNU project, but it adheres strongly to the UNIX ideal to do one thing well. It takes primarily flat text files and produces only 2D x-y plots of the data. It is designed to allow interactive operation and very simple batch scripted "programs."

Janert writes with the same philosophy, taking advantage of the interactive nature of Gnuplot to get the reader started producing plots immediately. By the end of the fourth chapter, I had everything I needed to create clean, simple two-variable plots that would easily serve the uses I have had in the past. The only thing I wish were offered earlier in the book is date and timestamp parsing. That had to wait for Chapter 8.

Gnuplot does have a number of shortcomings that result from this philosophical simplicity. All variables are global. The control structures and function definition syntax are rudimentary. The functionality that a user of modern scripting environments might expect (locally scoped variables?) just doesn't exist or is simulated with what can charitably be called hacks. Janert doesn't shy away from these limitations but, rather, explains the reasoning that has led to them and then shows how to make the most of what is there.

In the latter half of the book, Janert deals with scripting, streaming data, and even animation. These aren't things I ever expect to do with Gnuplot, but it's interesting to know that you can. In all of these areas his writing is to the point and clear. The next time I have a data series I need to visualize, I know now that I'll be able to get some quick clean plots.

Grokking Algorithms: An Illustrated Guide for Programmers and Other Curious People

Aditya Y. Bhargava

Manning Publications Co., 2016, 238 pages

ISBN 978-1-61729-223-1

Reviewed by Mark Lamourine

Grokking Algorithms is a breezy, comfortable introduction to computer algorithms. I was originally attracted to it because of the *grok* in the title, a term coined by Robert Heinlein in *Stranger in a Strange Land* and adopted by the computer community by the 1980s to mean *to understand deeply*. While I've heard and used the term, I've never before seen it in the title of a book and wondered whether any book could aspire to help someone grok anything. I was also intrigued by the subtitle, *An Illustrated Guide for Programmers and Other Curious People*. Are there non-programmers who are curious about algorithms, and how would one approach algorithms with them?

Bhargava is both the writer and illustrator. He writes and uses his drawings to clarify the text. His drawings feature a pen-and-ink style that is much softer than the typical spare computer-generated graphics so common today. This, along with his inclusive narrative style, gives his book a living-room feel that is in sharp contrast to most technical writing being done.

This isn't an academic tome—though it might be used as a high-school or freshman college intro to algorithms course—and Bhargava isn't a theorist; he's an artist by training and a practical coder by profession. His goal is to give the reader a sense of how algorithms are created and how they work. He offers one of the better explanations of Big O notation that I've seen. He also makes a point that I think is often missed, that two algorithms of the same order can still have very different efficiency if the cost of a single cycle is higher than the other.

There are algorithms here that didn't exist or were not in common use when I studied. Bhargava devotes a chapter each to "Dynamic Programming" and "Greedy Algorithms," and applies the K-Nearest Neighbors algorithm to a simple OCR problem. In the closing chapter, he just mentions 10 more algorithms, talking about what they are used for and why they are important. These include a one-page exposition of tree algorithms, a few pages on MapReduce, and a page each on SHA hashing, Diffie-Hellman key exchange, and linear programming.

Even where Bhargava goes into more depth about how specific algorithms work, he's not all that concerned with implementation details. It's especially true when he talks about hash tables and sorts. He does go into enough detail to help the reader understand when and how to use a hash table, but he explains, correctly, that most modern languages have some hash table feature, and that most coders will never have to implement a hash table.

Reading *Grokking Algorithms* won't make you a programmer, but I really like it as an introduction for someone who is curious about why and how people solve problems with software. I do think the motivated reader will go away with a deeper understanding of how computers are used to ask and answer questions.

For Fun and Profit: A History of the Free and Open Source Software Revolution

Christopher Tozzi

The MIT Press, 2017, 324 pages

ISBN 978-0-262-03647-4

Reviewed by Michele Nelson

I volunteered to read and review Christopher Tozzi's *For Fun and Profit* because of my interest in the history of computing and how the tools I use today came to be. It turns out that I have lived through a revolution without even realizing it.

In his lengthy introduction, Tozzi, an Assistant Professor of History at Howard University, offers the theory that what he refers to as the free and open source software (FOSS) revolution that began in the 1980s has much in common with the French Revolution of 1789. The French Revolution sought to achieve liberty, fraternity, and equality. The FOSS revolution sought to restore software freedom. Both changed the world.

I always thought that "free software" referred to any software you don't have to pay to use, and that "open source" meant software you don't have to pay to use that also has the source code available at no cost. According to Tozzi, however, FOSS is not so easy to define. He feels that both "free software" and "open source" are ambiguous terms. After much discussion in the introduction, he defines "free software," for the purposes of this book, as software "whose programmers or users call it such because its source code can be studied and modified freely by people who use the software, whether or not the source code costs money." He uses the term "open source" to refer to software "whose creators and users preferred that term over 'free software.'" That still seems a bit ambiguous to me.

The book is divided into six chapters, tracing the FOSS revolution from its beginnings to its current status. Tozzi begins with

a discussion of the origins of hacker culture, defining the word "hacker" as "the class of programmers who espouse the hacker ethic" and devoting several pages to the definition of the "hacker ethic." He goes on to tell the story of the birth of the BSD and GNU operating systems, devotes a chapter to "The Story of Linux," and also covers Richard Stallman's GNU Project and the Free Software Foundation in detail. His story continues with the "moderate phase" of the revolution—Linux and GNU distributions, office apps, email, and Web—Apache, Samba, MySQL, and PHP.

The chapter titled "The FOSS Revolutionary Wars" was the most interesting to me. According to the author, there were two wars being fought: one inside the FOSS community—"The FOSS Civil War"—and another that pitted the FOSS community against proprietary software companies, chiefly Microsoft.

Tozzi describes the battle within the FOSS community as a fight over what "free software" and "open source" actually meant. He concludes that this battle was not won by either side and is still a contentious issue for some. At the same time as this conflict was causing a rift in the community, there were external threats that required them to band together despite their disagreements. As Tozzi explains it, proprietary software companies, especially Microsoft, were getting increasingly nervous as FOSS products began to be accepted, even endorsed, in the business world. When Netscape released the Mozilla browser as an open source product in 1998, Microsoft went on the offensive. The story of that battle, involving leaked internal Microsoft documents, lawsuits and countersuits, and a report from an outfit called the Alexis de Tocqueville Institution contending that Linus Torvalds "must have plagiarized much of the Linux source code" from Andrew Tanenbaum's Minix operating system, is one of the best parts of this book.

In the final chapter, Tozzi describes how FOSS has evolved since the early 2000s, looking at some key developments: the endorsement of FOSS by many large companies, including Microsoft; the Android mobile OS; Ubuntu/Linux; and cloud and embedded computing. This chapter also includes a discussion of the "free culture" movement, citing Creative Commons and Wikipedia as two examples of projects that have "extended FOSS principles into new territory." He ends with a discussion of diversity, or rather the lack of it, in the software industry as a whole and in the FOSS community in particular.

In *For Fun and Profit*, Christopher Tozzi relates the history of the FOSS revolution from the origins of hacker culture through the end of the revolutionary wars to the present. Some of the chapter introductions are rather tedious, and the writing could be livelier, but overall, I found this to be an interesting read that I will no doubt revisit.

Statement of Ownership, Management, and Circulation, 10/01/2017

Title: ;login: Pub. No. 0008-334. Frequency: Quarterly. Number of issues published annually: 4. Subscription price \$90.

Office of publication: USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.

Headquarters of General Business Office of Publisher: Same. Publisher: Same.

Editor: Rik Farrow; Managing Editor: Michele Nelson, located at office of publication.

Owner: USENIX Association. Mailing address: As above.

Known bondholders, mortgagees, and other security holders owning or holding 1 percent or more of total amount of bonds, mortgages, or other securities: None.

The purpose, function, and nonprofit status of this organization and the exempt status for federal income tax purposes have not changed during the preceding 12 months.

Extent and Nature of Circulation		Average No. Copies Each Issue During Preceding 12 Months	No. Copies of Single Issue (Fall 2017) Published Nearest to Filing Date
a. Total Number of Copies		2587	2400
b. Paid Circulation	(1) Outside-County Mail Subscriptions	1192	1108
	(2) In-County Subscriptions	0	0
	(3) Other Non-USPS Paid Distribution	738	725
	(4) Other Classes	0	0
c. Total Paid Distribution		1930	1833
d. Free Distribution By Mail	(1) Outside-County	0	0
	(2) In-County	76	77
	(3) Other Classes Mailed Through the USPS	35	37
	(4) Free Distribution Outside the Mail	306	125
e. Total Free Distribution		417	239
f. Total Distribution		2347	2072
g. Copies Not Distributed		240	328
h. Total		2587	2400
i. Percent Paid		82%	88%
Paid Electronic Copies		512	490
Total Paid Print Copies		2442	2323
Total Print Distribution		2859	2562
Percent Paid (Both Print and Electronic Copies)		85%	91%

I certify that the statements made by me above are correct and complete.

Michele Nelson, Managing Editor

10/1/17