

---

# Algorithm Design and Applications

**Michael T. Goodrich**

Department of Information and Computer Science  
University of California, Irvine

**Roberto Tamassia**

Department of Computer Science  
Brown University

---

**WILEY**

VICE PRESIDENT & PUBLISHER	Don Fowley
EXECUTIVE EDITOR	Beth Lang Golub
EDITORIAL ASSISTANT	Jayne Ziemba
ASSISTANT MARKETING MANAGER	Debbie Martin
ASSOCIATE PRODUCTION MANAGER	Joyce Poh
PRODUCTION EDITOR	Wanqian Ye
COVER CREDIT	© Ansel Adams/U.S. National Archives and Records Administration

This book was set by the authors and printed and bound by Courier Kendallville.

This book is printed on acid free paper. ∞

Founded in 1807, John Wiley & Sons, Inc. has been a valued source of knowledge and understanding for more than 200 years, helping people around the world meet their needs and fulfill their aspirations. Our company is built on a foundation of principles that include responsibility to the communities we serve and where we live and work. In 2008, we launched a Corporate Citizenship Initiative, a global effort to address the environmental, social, economic, and ethical challenges we face in our business. Among the issues we are addressing are carbon impact, paper specifications and procurement, ethical conduct within our business and among our vendors, and community and charitable support. For more information, please visit our website: [www.wiley.com/go/citizenship](http://www.wiley.com/go/citizenship).

Copyright © 2015 John Wiley & Sons, Inc. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923 (Web site: [www.copyright.com](http://www.copyright.com)). Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030-5774, (201) 748-6011, fax (201) 748-6008, or online at: [www.wiley.com/go/permissions](http://www.wiley.com/go/permissions).

Evaluation copies are provided to qualified academics and professionals for review purposes only, for use in their courses during the next academic year. These copies are licensed and may not be sold or transferred to a third party. Upon completion of the review period, please return the evaluation copy to Wiley. Return instructions and a free of charge return shipping label are available at: [www.wiley.com/go/returnlabel](http://www.wiley.com/go/returnlabel). If you have chosen to adopt this textbook for use in your course, please accept this book as your complimentary desk copy. Outside of the United States, please contact your local sales representative..

***Library of Congress Cataloging in Publication Data:***

Goodrich, Michael T., author.  
 Algorithm design and applications / Michael T. Goodrich, Department of Information and Computer Science,  
 University of California, Irvine, Roberto Tamassia, Department of Computer Science, Brown University.  
 pages cm  
 Includes bibliographical references and index.  
 ISBN 978-1-118-33591-8 (hardback)  
 1. Computer algorithms. 2. Data structures (Computer science) I. Tamassia, Roberto, 1960- author. II. Title.  
 QA76.9.A43G668 2014  
 005.7'3--dc23

2014021534

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

### A More Realistic Analysis of Fisher-Yates Random Shuffling

In the analysis given earlier for the Fisher-Yates shuffling algorithm to generate a random permutation, we assumed that the  $\text{random}(k)$  method, which returns a random integer in the range  $[0, k - 1]$ , always runs in  $O(1)$  time. If we are using a random-number generator based on the use of unbiased random bits, however, this is not a realistic assumption.

In a system based on using random bits, we would most naturally implement the  $\text{random}(k)$  method by generating  $\lceil \log k \rceil$  random bits, interpreting these bits as an unsigned integer, and then repeating this operation until we got an integer in the range  $[0, k - 1]$ . Thus, counting each iteration in such an algorithm as a “step,” we could conservatively say that the  $\text{random}(k)$  method runs in 1 step with probability  $1/2$ , 2 steps with probability  $1/2^2$ , and, in general, in  $i$  steps with probability  $1/2^i$ . That is, its running time is a geometric random variable with parameter

$$p = \frac{1}{2}.$$

Under this more realistic assumption, the running time of the Fisher-Yates random permutation algorithm is proportional to the sum,

$$Y = Y_1 + Y_2 + \cdots + Y_{n-1},$$

where each  $Y_i$  is the number of steps performed in the  $i$ th call to the  $\text{random}$  method. In other words,  $Y$  is the sum of  $n - 1$  independent geometric random variables with parameter  $p = 1/2$ , since the running times of the calls to the  $\text{random}(k)$  method are independent. Thus, focusing on the steps used in calls to this method, the expected running time of the Fisher-Yates algorithm is

$$E[Y] = 2(n - 1).$$

Given this information, we can use the Chernoff bound for a sum of independent geometric random variables given in Theorem 19.15, with  $\alpha = 2$ , to bound the probability that the Fisher-Yates algorithm takes more than  $4n$  steps as follows:

$$\Pr(Y > 4n) \leq e^{-n/5}.$$

Therefore, under this more realistic analysis, the Fisher-Yates algorithm runs in  $O(n)$  time with high probability.

- R-19.12** Suppose we have a six-sided die, which we roll  $n$  times, and let  $X$  denote the number of times we roll a 1.
- (a) What is  $E[X]$ ?
  - (b) Show that  $X < n/3$  with high probability.
- R-19.13** Draw an example skip list resulting from performing the following sequence of operations on the skip list in Figure 19.18: `remove(38)`, `insert(8,x)`, `insert(24,y)`, `remove(55)`. Assume the coin flips for the first insertion yield two heads followed by tails, and those for the second insertion yield three heads followed by tails.
- R-19.14** Give a pseudocode description of the `remove` dictionary operation, assuming the dictionary is implemented by a skip-list structure.

---

## Creativity

- C-19.1** Suppose that Bob wants a constant-time method for implementing the `random( $k$ )` method, which returns a random integer in the range  $[0, k - 1]$ . Bob has a source of unbiased bits, so to implement `random( $k$ )`, he samples  $\lceil \log k \rceil$  of these bits, interprets them as an unsigned integer,  $K$ , and returns the value  $K \bmod k$ . Show that Bob's algorithm does not return every integer in the range  $[0, k - 1]$  with equal probability.
- C-19.2** Design a variation of Algorithm 19.2 (`randomSort`) that inserts the pairs  $(r_i, x_i)$  into a balanced tree and calls itself recursively when  $r_i$  is found to be equal to one of the previously generated random values. Give pseudocode for this variation of the algorithm and analyze its running time. Also, discuss its advantages and disadvantages with respect to the original algorithm.
- C-19.3** Design a variation of Algorithm 19.2 (`randomSort`) that begins by generating distinct random values,  $r_i$  ( $i = 1, \dots, n$ ) and then sorts the pairs  $(r_i, x_i)$ . Give pseudocode for this variation of the algorithm and analyze its running time. Also, discuss its advantages and disadvantages with respect to the original algorithm.
- C-19.4** Consider a modification of the Fisher-Yates random shuffling algorithm where we replace the call to `random( $k + 1$ )` with `random( $n$ )`, and take the for-loop down to 0, so that the algorithm now swaps each element with another element in the array, with each cell in the array having an equal likelihood of being the swap location. Show that this algorithm does not generate every permutation with equal probability.
- Hint:** Consider the case when  $n = 3$ .
- C-19.5** Suppose you have a collection,  $S$ , of  $n$  distinct items and you wish to select a random sample of these items of size exactly  $\lceil n^{1/2} \rceil$ . Describe an efficient method for selecting such a sample so that each element in  $S$  has an equal probability of being included in the sample.