



HASHING IN COMPUTER SCIENCE

Fifty Years of Slicing and Dicing



ALAN G. KONHEIM

 **WILEY**

HASHING IN COMPUTER SCIENCE

FIFTY YEARS OF SLICING AND DICING

Alan G. Konheim



 **WILEY**

JOHN WILEY & SONS, INC., PUBLICATION

Copyright © 2010 by John Wiley & Sons, Inc. All rights reserved

Published by John Wiley & Sons, Inc., Hoboken, New Jersey
Published simultaneously in Canada

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 Section 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, (978) 750-8400, fax (978) 750-4470, or on the web at 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, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permission>.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Cataloging-in-Publication Data

Konheim, Alan G., 1934–

Hashing in computer science : fifty years of slicing and dicing / Alan G. Konheim.
p. cm.

ISBN 978-0-470-34473-6

1. Hashing (Computer science) 2. Cryptography. 3. Data encryption (Computer science) 4. Computer security. I. Title.

QA76.9.H36K65 2010
005.8'2–dc22

2009052123

Printed in Singapore

10 9 8 7 6 5 4 3 2 1

HASHING IN COMPUTER SCIENCE

*To my grandchildren ...
Madelyn, David, Joshua, and Karlina*

0.1 WHAT IS HASHING

The *Merriam-Webster Dictionary* (1974) provides several definitions for *hash*¹:

- *noun*: an American food made by combining and compressing *chopped-up* leftovers—meat, eggs, and potatoes.
- *verb*: to chop into small pieces; make into hash; mince.
- *noun*: colloquial for *hashish* (or *hashish*), the resin collected from the flowers of the cannabis plant. The primary active substance is THC (tetrahydrocannabinol), although several other cannabinoids are known to occur. Hash is usually smoked in pipes, water pipes, joints, and hookahs, sometimes mixed with cannabis flowers or tobacco. It can also be eaten.

A search for *hashing* on *Google* yields 1,720,000 hits; of course, it doesn't come close to the 670,000,000 hits for *sex* or even the 129,000,000 hits for *money*, but then ...

Although hashing may be exhilarating and even intoxicating, this book deals with its applications in computer science relating to the processing of information that is both *old* (1953+) and *new* (1990+).

0.2 MY HASHING ROOTS

Nat Rochester was the software project manager for the IBM 701 machine; together with Gene M. Amdahl, Elaine M. McGraw (née Boehme), and Arthur L. Samuel, they considered a key-value-to-address machine for the 701 assembler. Several sources agree that Amdahl invented linear open addressing, a form of hashing (*aka* scatter storage), when confronted with the problem of collisions.

I was a Research Staff member in the Department of Mathematical Sciences at the IBM Thomas J. Watson Research Center in Yorktown Heights (New York) from

¹Denis Khotimsky, a former student, directed me to www.onin.com/hhh/hhhexpl.htm, which gives an entirely different meaning to hashing, as follows:

Hashing ... it's a mixture of athleticism and sociability, hedonism and hard work, a refreshing escape from the nine-to-five dweebs you're stuck with five days a week. Hashing is an exhilaratingly fun combination of running, orienteering, and partying, where bands of harriers and harriettes chase hares on eight-to-ten kilometer-long trails through town, country, and desert, all in search of exercise, camaraderie, and good times.

This type of hashing began in Kuala Lumpur, Malaysia, in 1938.

1960 to 1982. Sometime in 1964, I was asked by my manager to meet with Wes Peterson (1924–2009) to learn about his work on hashing. Wes was a former employee of and now consultant to IBM; in 1964, he was a faculty member at one of the University of Florida campuses. Wes used simulation to evaluate the performance of *linear probing*, an algorithm developed during the design of the IBM 701 machine. As a result of our conversation, I started to work with Benjamin Weiss, then a post-doc at IBM, now a Professor Emeritus at the Hebrew University in Jerusalem, to develop a combinatorial analysis of hashing with linear probing.

In a footnote on page 529 of *The Art of Computer Programming: Volume 3/Sorting and Searching* (Addison-Wesley Publishing Company, 1973), Professor Donald Knuth wrote that his running-time analysis of Algorithm L “was the first non-trivial algorithm I had ever analyzed satisfactorily. . . .” He also noted “that more than ten years would go by before the derivation got into print!” Algorithm L is central to the performance of *linear probing*, which is essentially the hashing algorithm proposed by Gene Amdahl. Ben Weiss and I found the formula $N_j \equiv (j + 1)^{(j-1)}$ (Theorem 13.2 is Knuth’s Algorithm L), which counts the number of hash sequences inserting j keys in a hash table of size $j + 1$ leaving the last hash table address unoccupied. First, we computed by hand the values N_j for $j = 2, 3, 4$ until we recognized the form of the algebraic expression $(j + 1)^{(j-1)}$, a technique that I have suggested to students in other contexts.

I contacted Wes again in 1997, where he was and remains today a Professor of Computer Science at the University of Hawaii. I inquired about a sabbatical leave there in 1999. My wife and I visited Wes and his wife Hiromi at their home in December 1997 while celebrating our 40th anniversary. Wes had just been awarded the Japan Prize, and both Petersons were both contemplating sabbatical leaves in Japan starting in January 1999. Timing is everything! I was hired by Wes and Hiromi to watch over their dog Koko; I was certainly the highest paid dog sitter in the Hawaiian Islands. Koko was a constant source of innocent merriment during my 5-month sabbatical; he curiously developed a significant taste for *Hebrew National* salami, and our acquiescing to his new found tastes was criticized by both Wes and Koko’s veterinarian. However, before leaving for Japan, Wes warned my wife that *Koko* might die before their return. The miraculous and curative powers of *Hebrew National* salami—*We Answer To a Higher Authority*—added 2 years to Koko’s life. We saw Wes again when we returned to Hawaii to celebrate our 50th anniversary in July 2007.

0.3 THE ORIGINAL APPLICATION OF HASHING (DATA STORAGE)

When data are stored and manipulated in a computer system, a mechanism must be provided to access the data easily. We view information in storage as composed of *records* of variable sizes located in the system storage media in a variety of regions, especially in dynamic file systems. A unique *key* is associated with and stored in each record. This key could be a name or a suitable number, but in general, any string can serve as a unique identifier. The key is used by the programs processing the data to locate—to SEARCH for the address of—the desired record. When a telephone directory is used to determine a telephone number, SEARCH is somewhat easier than the general instance of file management because the records composed

of the triple (Name Address TelephoneNumber) are stored *sorted* on the Name field.

The concept of hashing was first mentioned in a January 1953 internal IBM document by Hans Peter Luhn and several years later in an article by Arnold I. Dumey; the word *hashing* was first used in an article by Robert Morris in 1968.

There is considerable literature for the *old* hashing starting at least in 1963 and continuing until 2005. We describe various hashing methods in Chapters 7–15.

0.4 NEW APPLICATIONS FOR HASHING

Because technology evolves quickly, certain problems arise in many diverse new disciplines. Chapter 16 describes the work of Karp and Rabin. In a 1987 paper, they observed that hashing could be applied to the search of data for a particular string, for example, the word *bomb* in monitored e-mail messages.

The tracking of music performed on radio and television is part of agreements between the Music Performance Trust Fund and the recording industry. As part of this process, it is necessary to listen to the performance of a song on radio and to identify a song; is the performance by the Glenn Miller band or U2? Techniques for *audio fingerprinting*—the automated recognition of music—appeared in 2002 and will be described in Chapter 17.

The IBM Corporation introduced the Data Encryption Standard (DES) in 1973. It generated considerable criticism but resulted in important changes in the relationship among the government, the business world and the academic community. The Web offered services to about 300,000 hosts (processing systems) in 1990. Today, millions of users are connected by the Internet. The October 2006 Forrester Research report predicted that “Non-travel online retail revenues ... **e-commerce** ... on the Web will top the quarter-trillion-dollar mark by 2011.” This has produced, a new criminal activity called identity theft. Cryptography offered a possible solution to these problems.

Transactions on the Web require the following:

- *Secrecy* to hide credit card information transmitted from a buyer to a seller
- *User authentication* to verify to each party in a transaction the identity of the other party
- *Message authentication* to detect unauthorized changes in transmitted data

Cryptography has been ported to the world of e-commerce, and hashing is one of the constituents of the secrecy/authentication protocols for e-commerce used to protect and defend Web transactions. We describe the exciting current work on hashing in e-Commerce in Chapter 18.

Not only are the mom-and-pop stores on main street disappearing, but it seems that only the large-box chain stores may survive. The villain or savior is the Web, which provides a convenient and often cost-effective way of purchasing some items and/or services. Among the products easily distributed in this medium are music, video, software, books, and data.

Because information in its native form can easily be copied, distributed, or altered without the owners or the purchasers of the information being aware, their

unrestricted distribution poses serious issues. We show how hashing is used in Chapter 19 to inhibit these forms of misuse.

0.5 ABOUT THIS BOOK

This book will describe the basic applications of hashing, analyze hashing protocols, and illustrate the tools needed by a student to understand basic and fundamental problems in computer systems.

Why should students study hashing, and why this book? The purpose of higher education in computer science and mathematics is to prepare students for careers in industry, government, and the university. Computer science—like music and cooking—is not a spectator sport; it requires participation. We all learn by practice, attempting to solve new problems by adapting the solutions of old problems. This book focuses on mathematical methods in information management, but its structural techniques are applicable in many areas of science and technology.

The intended audience includes upper-division students in a computer science/engineering program. The prerequisites include skill in some programming language and elementary courses in discrete mathematics and probability theory. The book is divided into four parts.

Part I contains brief reviews of the relevant mathematical tools

Part II deals with the original hashing for data storage management

Part III examines the appearance of new applications of hashing ideas

Problems and solutions are included as well as an extensive bibliography which concludes the presentation

0.6 ACKNOWLEDGMENTS

I am in debt to many people who have helped and encouraged me in writing this book.

- My friend of 36 years, Dr. Raymond Pickholtz, Professor Emeritus at George Washington University, who visited UCSB several times, read all the chapters, and provided advice.
- His son, Mr Andrew Pickholtz *Esq.*, who read and added material on his dongle patent in Chapter 19.
- Dr. Michele Covell of Google who was kind enough to read and suggest changes to Chapter 17.
- Mr. Vlado Kitanovski of the Faculty of Electrical Engineering and Information Technologies (Ss Cyril and Methodius University—Skopje) who was kind enough to read and suggest changes to the section on watermarking of images in Chapter 19.
- Dr. Feng Hao of Thales e-Security (Cambridge, United Kingdom) and Professor Ross Anderson of the University of Cambridge Computer Laboratory for their comments on iris identification in Chapter 19.
- Professor Haim Wolfson of Tel-Aviv University who clarified my presentation on geometric hashing.
- My son Keith whose graphic skills were indispensable.

- Finally, to my wife Carol of more than 52 years, who continues to amaze me by her wide-ranging talents. I could not have undertaken this book without her encouragement, assistance, and advice.

It's what you learn after you know it all that counts, attributed to Harry S. Truman.

Anyone who stops learning is old, whether at twenty or eighty, Henry Ford.

Teaching is the highest form of understanding, Aristotle.

3.4:	Laplace's Method	47
3.5:	The Saddle Point Method	48
3.6:	When Will the Saddle Point Method Work?	51
3.7:	The Saddle Point Bounds	52
3.8:	Examples of Saddle Point Analysis	54
4.	Discrete Probability Theory	64
4.1:	The Origins of Probability Theory	64
4.2:	Chance Experiments, Sample Points, Spaces, and Events	64
4.3:	Random Variables	66
4.4:	Moments—Expectation and Variance	67
4.5:	The Birthday Paradox	68
4.6:	Conditional Probability and Independence	69
4.7:	The Law of Large Numbers (LLN)	71
4.8:	The Central Limit Theorem (CLT)	73
4.9:	Random Processes and Markov Chains	73
5.	Number Theory and Modern Algebra	77
5.1:	Prime Numbers	77
5.2:	Modular Arithmetic and the Euclidean Algorithm	78
5.3:	Modular Multiplication	80
5.4:	The Theorems of Fermat and Euler	81
5.5:	Fields and Extension Fields	82
5.6:	Factorization of Integers	86
5.7:	Testing Primality	88
6.	Basic Concepts of Cryptography	94
6.1:	The Lexicon of Cryptography	94
6.2:	Stream Ciphers	95
6.3:	Block Ciphers	95
6.4:	Secrecy Systems and Cryptanalysis	95
6.5:	Symmetric and Two-Key Cryptographic Systems	97
6.6:	The Appearance of Public Key Cryptographic systems	98
6.7:	A Multitude of Keys	102
6.8:	The RSA Cryptosystem	102
6.9:	Does PKC Solve the Problem of Key Distribution?	105
6.10:	Elliptic Groups Over the Reals	106
6.11:	Elliptic Groups Over the Field $\mathbb{Z}_{m,2}$	108
6.12:	Elliptic Group Cryptosystems	110
6.13:	The Menezes-Vanstone Elliptic Curve Cryptosystem	111
6.14:	Super-Singular Elliptic Curves	112
PART II:	HASHING FOR STORAGE: DATA MANAGEMENT	117
7.	Basic Concepts	119
7.1:	Overview of the Records Management Problem	119

CONTENTS

PREFACE	xiii
PART I: MATHEMATICAL PRELIMINARIES	1
1. Counting	3
1.1: The Sum and Product Rules	4
1.2: Mathematical Induction	5
1.3: Factorial	6
1.4: Binomial Coefficients	8
1.5: Multinomial Coefficients	10
1.6: Permutations	10
1.7: Combinations	14
1.8: The Principle of Inclusion-Exclusion	17
1.9: Partitions	18
1.10: Relations	19
1.11: Inverse Relations	20
Appendix 1: Summations Involving Binomial Coefficients	21
2. Recurrence and Generating Functions	23
2.1: Recursions	23
2.2: Generating Functions	24
2.3: Linear Constant Coefficient Recursions	25
2.4: Solving Homogeneous LCCRs Using Generating Functions	26
2.5: The Catalan Recursion	30
2.6: The Umbral Calculus	31
2.7: Exponential Generating Functions	32
2.8: Partitions of a Set: The Bell and Stirling Numbers	33
2.9: Rouché's Theorem and the Lagrange's Inversion Formula	37
3. Asymptotic Analysis	40
3.1: Growth Notation for Sequences	40
3.2: Asymptotic Sequences and Expansions	42
3.3: Saddle Points	45

7.2:	A Simple Storage Management Protocol: Plain Vanilla Chaining	120
7.3:	Record-Management with Sorted Keys	122
8.	Hash Functions	124
8.1:	The Origin of Hashing	124
8.2:	Hash Tables	126
8.3:	A Statistical Model for Hashing	129
8.4:	The Likelihood of Collisions	130
9.	Hashing Functions: Examples and Evaluation	132
9.1:	Overview: The Tradeoff of Randomization Versus Computational Simplicity	132
9.2:	Some Examples of Hashing Functions	132
9.3:	Performance of Hash Functions: Formulation	135
9.4:	The χ^2 -Test	137
9.5:	Testing a Hash Function	138
9.6:	The McKenzie et al. Results	139
10.	Record Chaining with Hash Tables	141
10.1:	Separate Chaining of Records	141
10.2:	Analysis of Separate Chaining Hashing Sequences and the Chains They Create	143
10.3:	A Combinatorial Analysis of Separate Chaining	148
10.4:	Coalesced Chaining	152
10.5:	The Pittel-Yu Analysis of EICH Coalesced Chaining	155
10.6:	To Separate or to Coalesce; and Which Version? That Is the Question	161
11.	Perfect Hashing	164
11.1:	Overview	164
11.2:	Chichelli's Construction	164
12.	The Uniform Hashing Model	167
12.1:	An Idealized Hashing Model	167
12.2:	The Asymptotics of Uniform Hashing	171
12.3:	Collision-Free Hashing	172
13.	Hashing with Linear Probing	174
13.1:	Formulation and Preliminaries	174
13.2:	Performance Measures for LP Hashing	176
13.3:	All Cells Other than HT_{n-1} in the Hash-Table of n Cells are Occupied	178
13.4:	m -Keys Hashed into a Hash Table of n Cells Leaving Cell HT_{n-1} Unoccupied	180
13.5:	The Probability Distribution for the Length of a Search	182

13.6:	Asymptotics	184
13.7:	Hashing with Linear Open Addressing: Coda	189
13.8:	A Possible Improvement to Linear Probing	189
14.	Double Hashing	205
14.1:	Formulation of Double Hashing	205
14.2:	Progressions and Strides	207
14.3:	The Number of Progressions Which Fill a Hash-Table Cell	208
14.3.1:	Progression Graphs	209
14.4:	Dominance	215
14.5:	Insertion-Cost Bounds Relating Uniform and Double Hashing	216
14.6:	UsuallyDoubleHash	218
14.7:	The UDH Chance Experiment and the Cost to Insert the Next Key by Double Hashing	219
14.8:	Proof of Equation (14.12a)	221
14.9:	UsuallyDoubleHash'	224
14.10:	Proof of Equation (14.12b)	225
15.	Optimum Hashing	227
15.1:	The Ullman–Yao Framework	227
15.1.1:	The Ullman–Yao Hashing Functions	227
15.1.2:	Ullman–Yao INSERT(k) and SEARCH(k)	228
15.1.3:	The Ullman–Yao Statistical Model	228
15.2:	The Rates at Which a Cell is Probed and Occupied	230
15.3:	Partitions of (i) Scenarios, (i) Subscenarios, and Their Skeletons	232
15.3.1:	(i) Subscenarios	234
15.3.2:	Skeletons	234
15.4:	Randomly Generated m -Scenarios	234
15.5:	Bounds on Random Sums	240
15.6:	Completing the Proof of Theorem 15.1	244
PART III:	SOME NOVEL APPLICATIONS OF HASHING	247
16.	Karp-Rabin String Searching	249
16.1:	Overview	249
16.2:	The Basic Karp-Rabin Hash-Fingerprint Algorithm	250
16.3:	The Plain Vanilla Karp-Rabin Fingerprint Algorithm	251
16.4:	Some Estimates on Prime Numbers	253
16.5:	The Cost of False Matches in the Plain Vanilla Karp-Rabin Fingerprint Algorithm	253
16.6:	Variations on the Plain Vanilla Karp-Rabin Fingerprint Algorithm	255
16.7:	A Nonhashing Karp-Rabin Fingerprint	255

17. Hashing Rock and Roll	259
17.1: Overview of Audio Fingerprinting	259
17.2: The Basics of Fingerprinting Music	260
17.3: Haar Wavelet Coding	262
17.4: Min-Hash	266
17.5: Some Commercial Fingerprinting Products	273
18. Hashing in E-Commerce	275
18.1: The Varied Applications of Cryptography	275
18.2: Authentication	276
18.3: The Need for Certificates	277
18.4: Cryptographic Hash Functions	278
18.5: X.509 Certificates and CCIT Standardization	280
18.6: The Secure Socket Layer (SSL)	281
18.7: Trust on the Web ... Trust No One Over 40!	282
18.8: MD5	284
18.9: Criticism of MD5	289
18.10: The Wang-Yu Collision Attack	291
18.11: Steven's Improvement to the Wang-Yu Collision Attack	291
18.12: The Chosen-Prefix Attack on MD5	292
18.13: The Rogue CA Attack Scenario	293
18.14: The Secure Hash Algorithms	294
18.15: Criticism of SHA-1	296
18.16: SHA-2	296
18.17: What Now?	298
Appendix 18: Sketch of the Steven's Chosen Prefix Attack	301
19. Hashing and the Secure Distribution of Digital Media	307
19.1: Overview	307
19.2: Intellectual Property (Copyrights and Patents)	308
19.3: Steganography	313
19.4: Boil, Boil, Toil ... and But First, Carefully Mix	314
19.5: Software Distribution Systems	314
19.6: Watermarks	315
19.7: An Image-Processing Technique for Watermarking	317
19.8: Using Geometric Hashing to Watermark Images	320
19.9: Biometrics and Hashing	323
19.10: The Dongle	329
Appendix 19: Reed-Solomon and Hadamard Coding	332
Exercises and Solutions	343
INDEX	382

PART I

MATHEMATICAL PRELIMINARIES

