Handbook of Algorithms and Data Structures

G.H. Gonnet



INTERNATIONAL COMPUTER SCIENCE SERIES

01850

• •

· · · · ·

Handbook of ALGORITHMS and DATA STRUCTURES

INTERNATIONAL COMPUTER SCIENCE SERIES

Consulting editors

A D McGettrick University of Strathclyde

J van Leeuwen University of Utrecht



Handbook of ALGORITHMS and DATA STRUCTURES

G. H. Gonnet

ADDISON-WESLEY PUBLISHING COMPANY London · Reading, Massachusetts · Menlo Park, California · Amsterdam Don Mills, Ontario · Manila · Singapore · Sydney · Tokyo

© 1984 Addison-Wesley Publishers Limited

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 or otherwise, without prior written permission of the publisher.

Set by the author on an APS-5 photosetter using output generated by troff, the UNIX text-processing system (UNIX is a trademark of Bell Laboratories)

Cover illustration produced by John Vince using PICASO, the computer graphics system at Middlesex Polytechnic

Printed in Finland by OTAVA. Member of Finnprint

British Library Cataloguing in Publication Data

Gonnet, G. H.
Handbook of algorithms and data structures.
1. Electronic digital computers—Programming
2. Algorithms
I. Title
001.6'42 QA76.6

ISBN 0-201-14218-X

Library of Congress Cataloging in Publication Data

Gonnet, G. H. (Gaston H.)

Handbook of algorithms and data structures. (International computer science series)
Bibliography: p. Includes index.
1. Electronic digital computers—Programming.
2. Algorithms. 3. Data structures (Computer science) 4 9794
1. Title. 11. Series.
OA76.6.G636 1984 001.64'2 83-301.

ISBN 0-201-14218-X

ABCDEF8987654

To my boys: Miguel, Pedro, Julio and Ignacio and my girls: Araina and Marte



BIBLIOTHEQUE DU CERIST

Preface

Computer Science has been, throughout its evolution, more an art than a science. My favourite example which illustrates this point is to compare a major software project (like the writing of a compiler) with any other major project (like the construction of the CN tower in Toronto). It would be absolutely unthinkable to let the tower fall down a few times while its design was being debugged: even worse would be to open it to the public before discovering some other fatal flaw. Yet this mode of operation is being used everyday by almost everybody in software production.

Presently it is very difficult to "stand on your predecessor's shoulders", most of the time we stand on our predecessor's toes, at best. This handbook was written with the intention of making available to the computer scientist, instructor or programmer the wealth of information which the field has generated in the last 20 years.

Most of the results are extracted from the given references. In some cases the author has completed or generalized some of these results. Accuracy is certainly one of our goals, and consequently the author will cheerfully pay \$2.00 for each first report of any type of error appearing in this handbook.

Many people helped me directly or indirectly to complete this project. Firstly I owe my family hundreds of hours of attention. All my students and colleagues had some impact. In particular I would like to thank M. C. Momard, N. Ziviani, J. I. Munro, P. A. Lanson, D. Rotem and D. Wood. Very special thanks go to F. W. Tompa who is also the coauthor of chapter 2. The source material for this chapter appears in a joint paper in the November 1983 *Communications of the ACM*.

Montevideo December 1983 G. H. Gonnet

BIBLIOTHEQUE DU CERIST

Contents

Preface			vn
Chapter 1	INTRO	DDUCTION	1
1.1	Struct	ure of the chapters	1
1.2	Namin	ig of variables	3
1.3	Proba	bilities	3
1.4	Asymptotic notation		
1.5	About	the programming languages	5
1.6	On the	e code for the algorithms	6
1.7	Comp	lexity measures and real timings	6
Chapter 2	BASIC	C CONCEPTS	8
2.1	Data s	tructure description	8
	2.1.1	Grammar for data objects	8
	2.1.2	Constraints for data objects	11
2.2	Algori	thm descriptions	12
	2.2.1	Basic (or atomic) operations	13
	2.2.2	Building procedures	14
	2.2.3	Interchangeability	21
Chapter 3	SEAR	CHING ALGORITHMS	23
3.1	Seque	ntial search	23
	3.1.1	Basic Sequential search	23
	3.1.2	Self-organizing Sequential search: Move to Front method	25
	3.1.3	Self-organizing Sequential search: Transpose method	28
	3.1.4	Optimal Sequential search	- 30
	3.1.5	Jump search	- 30
3.2	Sorted array search		31
	3.2.1	Binary search	32
	3.2.2	Interpolation search	34
	3.2.3	Interpolation-Sequential search	35
3.3	Hashir	ıg	37
	3.3.1	Uniform probing hashing	40
	3.3.2	Random probing hashing	41
	3.3.3	Linear probing hashing	42
	3.3.4	Double hashing	45
	3.3.5	Quadratic hashing	47

x HANDBOOK OF ALGORITHMS AND DATA STRUCTURES

	3.3.6 Ordered baching	.10
	3.3.7 Reorganization for Uniform probing: Brent's algorithm	51
	3.3.8 Reorganization for Uniform probing: Binary tree bashing	53
	3.3.9 Optimal bashing	55
	3.3.10 Direct chaining hashing	56
	3.3.11 Separate chaining hashing	59
	3.3.12 Coalesced hashing	60
	3.3.13 Extendible hashing	63
	3.3.14 Linear hashing	65
	3.3.15 External hashing using minimal internal storage	67
3.4	Recursive structures search	69
	3.4.1 Binary tree search	69
	3.4.2 B-trees	90
	3.4.3 Index and indexed sequential files	101
	3.4.4 Digital trees	103
3.5	Multidimensional search	112
	3.5.1 Quad trees	112
	3.5.2 K-dimensional trees	114
Chanter 4	SORTING ALGORITHMS	118
4.1	Techniques for sorting arrays	118
	4.1.1 Bubble sort	118
	4.1.2 Linear insertion sort	120
	4.1.3 Quicksort	121
	4.1.4 Shellsoft	124
	4.1.5 Heapsort	120
	4.1.6 Interpolation sort	128
	4.1.7 Linear probing sort	120
1.2	4.1.8 Summary	1.52
4.2	Sorting other data structures	100
	4.2.1 Merge sort	134
	4.2.2 Quicksoft for fists	1.20
	4.2.5 DUCKUI SOIL 4.2.4 Dodiw cort	120
	4.2.5 Hybrid methods of sorting	1.07
	4.2.6 Transart	1/12
13	Morging	144
4.9	4.3.1 List merging	144
	4.3.1 Eist merging	145
	4.3.3 Minimal-comparison merging	146
4.4	External sorting	147
	4.4.1 Balanced merge sort	152
	4.4.2 Cascade merge sort	153
	4.4.3 Polyphase merge sort	155
	4.4.4 Oscillating merge sort	158
	4.4.5 External Quicksort	160
	-	
Chanter 5	SELECTION ALCORITHMS	163
Unapter 3		105

5.1	Priority queues		
	5.1.1 Sorted/unsorted lists	164	

	5.1.2 P-trees	166
	5.1.3 Heaps	169
	5.1.4 VanEmde-Boas priority queues	172
	5.1.5 Pagodas	174
	5.1.6 Binary trees used as priority queues	177
	5.1.7 Binomial queues	181
	5.1.8 Summary	182
5.2	Selection of k^{th}	183
	5.2.1 Selection by sorting	183
	5.2.2 Selection by tail recursion	185
	5.2.3 Selection of the mode	186
Chapter 6	ARITHMETIC ALGORITHMS	188
6.1	Basic operations, multiplication/division	188
6.2	Other arithmetic functions	193
	6.2.1 Binary powering	193
	6.2.2 Arithmetic-geometric mean	195
	6.2.3 Transcendental functions	196
6.3	Matrix multiplication	198
	6.3.1 Strassen's matrix multiplication	199
	6.3.2 Further asymptotic improvements	200
6.4	Polynomial evaluation	201
Appendix I	Distributions derived from empirical observation	203
Appendix II	Asymptotic expansions	209
Appendix III	References	218
Appendix IV	Algorithms coded in Pascal and in C	254
Index		279