A Guide to Experimental Algorithmics

Computational experiments on algorithms can supplement theoretical analysis by showing what algorithms, implementations, and speed-up methods work best for specific machines or problems. This book guides the reader through the nuts and bolts of the major experimental questions: What should I measure? What inputs should I test? How do I analyze the data?

Answers to these questions draw on ideas from algorithm design and analysis, operating systems and memory hierarchies, and statistics and data analysis. The wide-ranging discussion includes a tutorial on system clocks and CPU timers, a survey of strategies for tuning algorithms and data structures, a cookbook of methods for generating random combinatorial inputs, and a demonstration of variance reduction techniques. Numerous case studies and examples show how to apply these concepts.

All the necessary concepts in computer architecture and data analysis are covered so that the book can be used by anyone who has taken a course or two in data structures and algorithms. A companion website, AlgLabs (www.cs.amherst.edu/alglab) contains downloadable files, programs, and tools for use in projects.

Dr. Catherine C. McGeoch is the Beitzel Professor of Technology and Society in the Department of Computer Science at Amherst College. Professor McGeoch was co-founder (with David S. Johnson) in 1990 of the Discrete Mathematics and Theoretical Computer Science (DIMACS) Implementation Challenges. In 1999 she co-founded (with Michael Goodrich) the annual Workshop on Algorithm Engineering and Experimentation (ALENEX), sponsored by SIAM. She was Editor-in-Chief of the ACM Journal of Experimental Algorithmics from 2003 to 2008 and currently serves on the ACM Publications Board.
A Guide to Experimental Algorithmics

CATHERINE C. MCGEEOCH

Amherst College
To my parents, Jim and Nancy Cole
Contents

Preface

1 Introduction
   1.1 Why Do Experiments? 2
   1.2 Key Concepts 6
   1.3 What’s in the Book? 11
   1.4 Chapter Notes 12
   1.5 Problems and Projects 14

2 A Plan of Attack
   2.1 Experimental Goals 20
   2.2 Experimental Design Basics 25
   2.3 Chapter Notes 45
   2.4 Problems and Projects 47

3 What to Measure
   3.1 Time Performance 50
   3.2 Solution Quality 83
   3.3 Chapter Notes 94
   3.4 Problems and Projects 95

4 Tuning Algorithms, Tuning Code
   4.1 Reducing Instruction Counts 100
   4.2 Tuning to Reduce Instruction Costs 135
   4.3 The Tuning Process 145
   4.4 Chapter Notes 147
   4.5 Problems and Projects 149

5 The Toolbox
   5.1 The Test Program 154
## Contents

- Chapter 5 Generating Random Inputs 161
- Chapter 6 Creating Analysis-Friendly Data 181
- Chapter 7 Data Analysis 215
- Index 257

### Chapter 5
- Section 5.2 Generating Random Inputs 161
- Section 5.3 Chapter Notes 177
- Section 5.4 Problems and Projects 178

### Chapter 6
- Section 6.1 Variance Reduction Techniques 184
- Section 6.2 Simulation Shortcuts 204
- Section 6.3 Chapter Notes 211
- Section 6.4 Problems and Projects 212

### Chapter 7
- Section 7.1 Univariate Data 218
- Section 7.2 Bivariate Data: Correlation and Comparison 236
- Section 7.3 Understanding Y as a Function of X 240
- Section 7.4 Chapter Notes 252
Preface

This guidebook is written for anyone – student, researcher, or practitioner – who wants to carry out computational experiments on algorithms (and programs) that yield correct, general, informative, and useful results. (We take the wide view and use the term “algorithm” to mean “algorithm or program” from here on.)

Whether the goal is to predict algorithm performance or to build faster and better algorithms, the experiment-driven methodology outlined in these chapters provides insights into performance that cannot be obtained by purely abstract means or by simple runtime measurements. The past few decades have seen considerable developments in this approach to algorithm design and analysis, both in terms of number of participants and in methodological sophistication.

In this book I have tried to present a snapshot of the state-of-the-art in this field (which is known as experimental algorithmics and empirical algorithmics), at a level suitable for the newcomer to computational experiments. The book is aimed at a reader with some undergraduate computer science experience: you should know how to program, and ideally you have had at least one course in data structures and algorithm analysis. Otherwise, no previous experience is assumed regarding the other topics addressed here, which range widely from architectures and operating systems, to probability theory, to techniques of statistics and data analysis.

A note to academics: The book takes a nuts-and-bolts approach that would be suitable as a main or supplementary text in a seminar-style course on advanced algorithms, experimental algorithmics, algorithm engineering, or experimental methods in computer science. Several case studies are presented throughout; a companion website called AlgLab – Open Laboratory for Experiments on Algorithms makes the files, programs, and tools described in the case studies available for downloading. Suggestions for experimental problems and projects appear at the end of each chapter.
Preface

This book wouldn’t exist without the “number of participants” alluded to earlier, members of the research community who have worked to develop this new methodology while contributing a huge body of experiment-based research on design and analysis of algorithms, data structures, heuristics, and models of computation. I am grateful for all those collegial conversations during break-out sessions, carried out over countless cups of coffee: thanks to David Bader, Giuseppe Italiano, David S. Johnson, Richard Ladner, Peter Sanders, Matt Stallmann, and Cliff Stein. A huge thank you, especially, to Jon Bentley, whose comments, story ideas, and criticisms of draft versions of this book were immensely valuable. My editor Lauren Cowles also did a magnificent job of helping me to untangle knots in the draft manuscript.

Possibly more important to the final product than colleagues and readers are the family and friends who remind me that life is more than an endless bookwriting process: to Alex and Ian, Ruth and Stephen, Susan Landau, and Maia Ginsburg, thank you for keeping me sane.

And finally, very special thanks to the guy who fits all of the above categories and more: colleague, technical adviser, reader, supporter, husband, and friend. Thank you Lyle.

Catherine C. McGeoch
Amherst, Massachusetts
July 2011