Basic Concepts in Data Structures

Basic Concepts in Data Structures acquaints the reader with the theoretical side of the art of writing computer programs. Instead of concentrating on the technical aspects of how to instruct a computer to perform a certain task, the book switches to the more challenging question of what in fact should be done to solve a given problem.

The volume is the result of several decades of teaching experience in data structures and algorithms. It is self-contained and does not assume any prior knowledge other than of some basic programming and mathematical tools. Klein reproduces his oral teaching style in writing, with one topic leading to another, related one. Most of the classic data structures are covered, though not in a comprehensive manner. Alternatively, some more advanced topics, related to pattern matching and coding, are mentioned.

SHMUEL TOMI KLEIN started teaching in high school, repeating to his classmates almost daily the lectures of their mathematics teacher. As a computer science undergraduate at the Hebrew University of Jerusalem, he acted as teaching assistant in the Statistics Department and has since given courses and lectures on data structures, algorithms, and related topics in English, French, German, and Hebrew.

Klein’s research focuses on data compression and text-processing algorithms. He is a full professor and former chair of the Computer Science Department at Bar-Ilan University and a coauthor of more than 100 academic publications and 10 patents.
dedicated to

my spouse Rina

and our children Shoshanit and Itay
Avital and Ariel
Raanan and Yifat
Ayal and Yahav
Basic Concepts in Data Structures

SHMUEL TOMI KLEIN

Bar-Ilan University, Israel
Contents

List of Background Concepts  ix
Preface xi

1 Why Data Structures? A Motivating Example 1
  1.1 Boyer and Moore’s Algorithm 3
  1.2 The Bad-Character Heuristic 4
  1.3 The Good-Suffix Heuristic 7
    Exercises 12

2 Linear Lists 14
  2.1 Managing Data Storage 14
  2.2 Queues 16
  2.3 Stacks 21
  2.4 Other Linear Lists 28
    Exercises 31

3 Graphs 33
  3.1 Extending the Relationships between Records 33
  3.2 Graph Representations 38
  3.3 Graph Exploration 40
  3.4 The Usefulness of Graphs 41
    Exercises 47

4 Trees 50
  4.1 Allowing Multiple Successors 50
  4.2 General versus Binary Trees 52
  4.3 Binary Trees: Properties and Examples 55
  4.4 Binary Search Trees 58
    Exercises 64
5 AVL Trees
5.1 Bounding the Depth of Trees 65
5.2 Depth of AVL Trees 66
5.3 Insertions into AVL Trees 71
5.4 Deletions from AVL Trees 77
5.5 Alternatives 80
Exercises 80

6 B-Trees
6.1 Higher-Order Search Trees 83
6.2 Definition of B-Trees 85
6.3 Insertion into B-Trees 86
6.4 Deletions from B-Trees 90
6.5 Variants 92
Exercises 99

7 Heaps
7.1 Priority Queues 101
7.2 Definition and Updates 102
7.3 Array Implementation of Heaps 105
7.4 Construction of Heaps 106
7.5 Heapsort 110
Exercises 112

8 Sets
8.1 Representing a Set by a Bitmap 114
8.2 Union-Find 117
Exercises 125

9 Hash Tables
9.1 Calculating instead of Comparing 127
9.2 Hash Functions 129
9.3 Handling Collisions 134
9.4 Analysis of Uniform Hashing 140
9.5 Deletions from Hash Tables 147
9.6 Concluding Remarks 148
Exercises 150

10 Sorting
10.1 A Sequence of Sorting Algorithms 152
10.2 Lower Bound on the Worst Case 156
10.3 Lower Bound on the Average 160
Contents

10.4 Quicksort 166
10.5 Finding the $k$th Largest Element 170
   Exercises 177

11 Codes 178
11.1 Representing the Data 178
11.2 Compression Codes 179
11.3 Universal Codes 189
11.4 Error Correcting Codes 194
11.5 Cryptographic Codes 199
   Exercises 202

Appendix Solutions to Selected Exercises 205

Index 217
List of Background Concepts

Binary Search  page 15
Summing the $m$ First Integers  19
Asymptotic Notation  20
Mergesort  26
Number of Binary Trees  53
Depth of a Tree  65
Proving Properties of Trees  67
Fibonacci Sequence  70
Swapping Two Elements  108
Prime Numbers  131
Modular Arithmetic  133
Birthday Paradox  135
Infinite and Finite Summations  142
Approximating a Sum by an Integral  144
Average Complexity of an Algorithm  161
Recurrence Relations  175
Preface

After having mastered some high-level programming language and acquired knowledge in basic mathematics, it is time for a shift of attention. Instead of concentrating on the technical aspects of how to instruct a computer to perform a certain task, we switch to the more challenging question of what in fact should be done to solve a given problem. The aim of this book on data structures is to start acquainting the reader with the theoretical side of the art of writing computer programs. This may be considered as a first step in getting familiar with a series of similar fields, such as algorithms, complexity, and computability, that should be learned in parallel to improve practical programming skills.

The book is the result of several decades of teaching experience in data structures and algorithms. In particular, I have taught a course on Data Structures more than 30 times. The book is self-contained and does not assume any prior knowledge of data structures, just a comprehension of basic programming and mathematics tools generally learned at the very beginning of computer science or other related studies. In my university, the course is given in the second semester of the first year of the BSc program, with a prerequisite of Discrete Mathematics and Introduction to Programming, which are first-semester courses. The format is two hours of lecture plus two hours of exercises, led by a teaching assistant, per week.

I have tried to reproduce my oral teaching style in writing. I believe in associative learning, in which one topic leads to another, related one. Although this may divert attention from the central, currently treated subject, it is the cumulative impact of an entire section or chapter that matters. There was no intention to produce a comprehensive compendium of all there is to know about data structures but rather to provide a collection of what many could agree to be its basic ingredients and major building blocks, on which subsequent courses on algorithms could rely. In addition, many more advanced topics are mentioned.
Each chapter comes with its own set of exercises, many of which have appeared in written exams. Solutions to selected exercises appear in the appendix. There are short inserts treating some background concepts: they are slightly indented, set in another font, and separated from the main text by rules. Though each chapter could be understood on its own, even if it has pointers to earlier material, the book has been written with the intent of being read sequentially.

There is of course a long list of people to whom I am indebted for this project, and it is not possible to mention them all. Foremost, I owe all I know to the continuous efforts of my late father to offer me, from childhood on, the best possible education in every domain. This included also private lessons, and I am grateful to my teacher R. Gedalya Stein, who interspersed his Talmud lessons with short flashes to notions of grammar, history, and more, and thereby planted the seeds of the associative learning techniques I adopted later. There is no doubt that my high school mathematics teacher Fernand Biendel was one of the best; he taught us rigor and deep understanding, and the fact that more than half of our class ended up with a PhD in mathematics should be credited to him.

I wish to thank all my teachers at the Hebrew University of Jerusalem and at the Weizmann Institute of Science in Rehovot as well as my colleagues at Bar-Ilan University and elsewhere. Many of them had an impact on my academic career, especially the advisors for my theses, Eli Shamir and Aviezri Fraenkel. Amihood Amir is directly responsible for this book because he asked me, when he was department chair, to teach the course on Data Structures. Thanks also to Franya Franek for providing a contact at Cambridge University Press.

Last, but not least, I wish to thank my spouse and children, to whom this book is dedicated, for their ongoing encouragement and constructive comments during the whole writing period. As to my grandchildren, they have no idea what this is all about, so I thank them for just being there and lighting up my days with their love.