

1 Overview . . .	1
What Are Data Structures and Algorithms? . . . . .	1
Overview of Data Structures. . . . .	4
Overview of Algorithms. . . . .	6
Some Definitions. . . . .	6
Programming in Python. . . . .	8
Object-Oriented Programming. . . . .	23
Summary. . . . .	27
Questions. . . . .	27
Experiments. . . . .	28
2 Arrays . . .	29
The Array Visualization Tool. . . . .	30
Using Python Lists to Implement the Array Class. . . . .	37
The OrderedArray Visualization Tool. . . . .	47
Binary Search. . . . .	48
Python Code for an OrderedArray Class. . . . .	52
Logarithms. . . . .	58
Storing Objects. . . . .	60
Big O Notation. . . . .	65
Why Not Use Arrays for Everything? . . . . .	69
Summary. . . . .	69
Questions. . . . .	70
Experiments. . . . .	72
Programming Projects. . . . .	73
3 Simple Sorting . . .	75
How Would You Do It? . . . . .	76
Bubble Sort. . . . .	77
Selection Sort. . . . .	83
Insertion Sort. . . . .	87
Comparing the Simple Sorts. . . . .	96
Summary. . . . .	98
Questions. . . . .	98
Experiments. . . . .	100
Programming Projects. . . . .	101
4 Stacks and Queues . . .	103
Different Structures for Different Use Cases. . . . .	103
Stacks. . . . .	104
Queues. . . . .	116
Priority Queues. . . . .	126
Parsing Arithmetic Expressions. . . . .	132
Summary. . . . .	151
Questions. . . . .	152
Experiments. . . . .	154
Programming Projects. . . . .	155
5 Linked Lists . . .	157
Links. . . . .	158
The LinkedList Visualization Tool. . . . .	164
A Simple Linked List. . . . .	167

Double-Ended Lists.. . . . .	177
Linked List Efficiency.. . . . .	183
Abstract Data Types and Objects. . . . .	184
Ordered Lists.. . . . .	192
Doubly Linked Lists.. . . . .	198
Insertion and Deletion at the Ends.. . . . .	201
Insertion and Deletion in the Middle.. . . . .	204
Doubly Linked List as Basis for Deques.. . . . .	208
Circular Lists.. . . . .	209
Iterators.. . . . .	211
Summary.. . . . .	222
Questions.. . . . .	224
Experiments.. . . . .	226
Programming Projects.. . . . .	227
6 Recursion . . . . .	229
Triangular Numbers.. . . . .	230
Factorials. . . . .	237
Anagrams.. . . . .	239
A Recursive Binary Search.. . . . .	242
The Tower of Hanoi.. . . . .	245
Sorting with mergesort.. . . . .	255
Eliminating Recursion.. . . . .	267
Some Interesting Recursive Applications.. . . . .	275
Summary.. . . . .	280
Questions.. . . . .	281
Experiments.. . . . .	283
Programming Projects.. . . . .	283
7 Advanced Sorting . . . . .	285
Shellsort.. . . . .	285
Partitioning.. . . . .	294
Quicksort.. . . . .	302
Radix Sort.. . . . .	320
Timsort.. . . . .	324
Summary.. . . . .	327
Questions.. . . . .	329
Experiments.. . . . .	331
Programming Projects.. . . . .	332
8 Binary Trees . . . . .	335
Why Use Binary Trees?.. . . . .	335
Tree Terminology.. . . . .	337
An Analogy.. . . . .	340
How Do Binary Search Trees Work?.. . . . .	341
Finding a Node.. . . . .	346
Inserting a Node.. . . . .	350
Traversing the Tree.. . . . .	353
Finding Minimum and Maximum Key Values. . . . .	365
Deleting a Node.. . . . .	366
The Efficiency of Binary Search Trees.. . . . .	375

Trees Represented as Arrays.....	377
Printing Trees.....	379
Duplicate Keys.....	381
The BinarySearchTreeTester.py Program.....	382
The Huffman Code.....	386
Summary.....	393
Questions.....	394
Experiments.....	396
Programming Projects.....	397
9 2-3-4 Trees and External Storage . . .	401
Introduction to 2-3-4 Trees.....	401
The Tree234 Visualization Tool.....	408
Python Code for a 2-3-4 Tree.....	412
Efficiency of 2-3-4 Trees.....	430
2-3 Trees.....	432
External Storage.....	438
Summary.....	456
Questions.....	458
Experiments.....	459
Programming Projects.....	460
10 AVL and Red-Black Trees 463 Our Approach to the Discussion.. .	463
Balanced and Unbalanced Trees.....	464
AVL Trees.....	470
The Efficiency of AVL Trees.....	486
Red-Black Trees.....	487
Using the Red-Black Tree Visualization Tool.....	489
Experimenting with the Visualization Tool.....	492
Rotations in Red-Black Trees.....	497
Inserting a New Node.....	498
Deletion.....	508
The Efficiency of Red-Black Trees.....	509
2-3-4 Trees and Red-Black Trees.....	510
Red-Black Tree Implementation.....	514
Summary.....	515
Questions.....	517
Experiments.....	520
Programming Projects.....	521
11 Hash Tables 525 Introduction to Hashing.....	526
Open Addressing.....	536
Separate Chaining.....	565
Hash Functions.....	575
Hashing Efficiency.....	581
Hashing and External Storage.....	588
Summary.....	590
Questions.....	592
Experiments.....	594
Programming Projects.....	595
12 Spatial Data Structures 597 Spatial Data.....	597

Computing Distances Between Points.....	599
Circles and Bounding Boxes.....	601
Searching Spatial Data.....	611
Lists of Points.....	612
Grids.....	617
Quadtrees.....	633
Theoretical Performance and Optimizations.....	656
Practical Considerations.....	656
Further Extensions.....	658
Summary.....	659
Questions.....	661
Experiments.....	662
Programming Projects.....	663
13 Heaps 665 Introduction to Heaps.....	666
The Heap Visualization Tool.....	674
Python Code for Heaps.....	677
A Tree-Based Heap.....	684
Heapsort.....	686
Order Statistics.....	694
Summary.....	700
Questions.....	701
Experiments.....	703
Programming Projects.....	703
14 Graphs 705 Introduction to Graphs.....	705
Traversal and Search.....	718
Minimum Spanning Trees.....	734
Topological Sorting.....	740
Connectivity in Directed Graphs.....	753
Summary.....	759
Questions.....	760
Experiments.....	762
Programming Projects.....	763
15 Weighted Graphs . . .	767
Minimum Spanning Tree with Weighted Graphs.....	767
The All-Pairs Shortest-Path Problem.....	797
Efficiency.....	800
Intractable Problems.....	801
Summary.....	805
Questions.....	806
Experiments.....	808
Programming Projects.....	809
16 What to Use and Why 813 Analyzing the Problem.....	814
Foundational Data Structures.....	818
Special-Ordering Data Structures.....	824
Sorting.....	826
Specialty Data Structures.....	828
External Storage.....	829
Onward.....	831