Table of Contents

Preface xvii

Chapter 1: Python Data Types and Structures 1

Introducing Python 3.10 2

Installing Python 2

Windows operating system • 2

Linux-based operating systems • 3

Mac operating system • 3

Setting up a Python development environment 3

Setup via the command line • 3

Setup via Jupyter Notebook • 4

Overview of data types and objects 5

Basic data types 7

Numeric • 7

Boolean • 8

Sequences • 9

Strings • 9

Range • 10

Lists • 11

Membership, identity, and logical operations • 15

Membership operators • 15

Identity operators • 16

Logical operators • 17

Tuples • 18

Complex data types 19

Dictionaries • 19

Sets • 23

Immutable sets • 26

Python's collections module 27

Named tuples • 27

Deque • 28

Ordered dictionaries • 29

Default dictionary • 29

ChainMap object • 30

Counter objects • 31

UserDict • 32

UserList • 32

UserString • 33

Summary 33

Chapter 2: Introduction to Algorithm Design 35

Introducing algorithms 35

Performance analysis of an algorithm 38

Time complexity • 38

Space complexity • 40

Asymptotic notation 41

Theta notation • 42

Big O notation • 44

Omega notation • 47

Amortized analysis 49

Composing complexity classes 50

Computing the running time complexity of an algorithm 52

Summary 54

Exercises 55

Chapter 3: Algorithm Design Techniques and Strategies 57

Algorithm design techniques 58

Recursion 59

Divide and conquer 60

Binary search • 61

Merge sort • 63

Dynamic programming 68

Calculating the Fibonacci series • 70

Greedy algorithms 74

Shortest path problem • 76

Summary 89

Exercises 90

Chapter 4: Linked Lists 93

Arrays 94

Introducing linked lists 95

Nodes and pointers • 95

Singly linked lists 98

Creating and traversing • 98

Improving list creation and traversal • 99

Appending items • 100

Appending items to the end of a list • 100

Appending items at intermediate positions • 103

Querying a list • 106

Searching an element in a list • 107

Getting the size of the list • 107

Deleting items • 108

Deleting the node at the beginning of the singly linked list • 108

Deleting the node at the end in the singly linked list • 109

Deleting any intermediate node in a singly linked list • 111

Clearing a list • 113

Doubly linked lists 114

Creating and traversing • 115

Appending items • 116

Inserting a node at beginning of the list • 116

Inserting a node at the end of the list • 119

Inserting a node at an intermediate position in the list • 121

Querying a list • 123

Deleting items • 124

Circular lists 129

Creating and traversing • 131

Appending items • 131

Querying a list • 134

Deleting an element in a circular list • 134

Practical applications of linked lists 138

Summary 139

Exercise 140

Chapter 5: Stacks and Queues 141

Stacks 141

Stack implementation using arrays • 145

Stack implementation using linked lists • 148

Push operation • 149

Pop operation • 151

Peek operation • 154

Applications of stacks • 154

Queues 157

Python’s list-based queues • 159

The enqueue operation • 159

The dequeue operation • 161

Linked list based queues • 163

The enqueue operation • 163

The dequeue operation • 165

Stack-based queues • 166

Approach 1: When the dequeue operation is costly • 166

Approach 2: When the enqueue operation is costly • 168

Enqueue operation • 170

Dequeue operation • 170

Applications of queues • 173

Summary 176

Exercises 177

Chapter 6: Trees 179

Terminology 179

Binary trees 181

Implementation of tree nodes • 184

Tree traversal • 186

In-order traversal • 186

Pre-order traversal • 188

Post-order traversal • 190

Level-order traversal • 191

Expression trees • 194

Parsing a reverse Polish expression • 196

Binary search trees 201

Binary search tree operations • 202

Inserting nodes • 203

Searching the tree • 208

Deleting nodes • 209

Finding the minimum and maximum nodes • 215

Benefits of a binary search tree • 216

Summary 219

Exercises 219

Chapter 7: Heaps and Priority Queues 221

Heaps 221

Insert operation • 224

Delete operation • 229

Deleting an element at a specific location from a heap • 234

Heap sort • 236

Priority queues 237

Summary 244

Exercises 244

Chapter 8: Hash Tables 247

Introducing hash tables 248

Hashing functions • 249

Perfect hashing functions • 251

Resolving collisions 252

Open addressing • 254

Linear probing • 254

Implementing hash tables 256

Storing elements in a hash table • 257

Growing a hash table • 258

Retrieving elements from the hash table • 260

Testing the hash table • 262

Implementing a hash table as a dictionary • 263

Quadratic probing • 264

Double hashing • 267

Separate chaining • 272

Symbol tables 278

Summary 279

Exercise 279

Chapter 9: Graphs and Algorithms 281

Graphs 281

Directed and undirected graphs • 283

Directed acyclic graphs • 284

Weighted graphs • 285

Bipartite graphs • 285

Graph representations 286

Adjacency lists • 287

Adjacency matrix • 288

Graph traversals 291

Breadth-first traversal • 291

Depth-first search • 299

Other useful graph methods 305

Minimum Spanning Tree • 305

Kruskal's Minimum Spanning Tree algorithm • 306

Prim's Minimum Spanning Tree algorithm • 309

Summary 312

Exercises 312

Chapter 10: Searching 313

Introduction to searching 313

Linear search 314

Unordered linear search • 315

Ordered linear search • 317

Jump search 320

Binary search 325

Interpolation search 331

Exponential search 337

Choosing a search algorithm 341

Summary 342

Exercise 342

Chapter 11: Sorting 345

Technical requirements 345
Sorting algorithms 345
Bubble sort algorithms 346
Insertion sort algorithm 352
Selection sort algorithm 356
Quicksort algorithm 359
Implementation of quicksort 364
Timsort algorithm 369
Summary 374
Exercise 374

Chapter 12: Selection Algorithms 377

Technical requirements 377
Selection by sorting 378
Randomized selection 378
Quickselect • 379
Deterministic selection 383
Implementation of the deterministic selection algorithm • 386
Summary 393
Exercise 393

Chapter 13: String Matching Algorithms 395

Technical requirements 395
String notations and concepts 395
Pattern matching algorithms 397
The brute force algorithm 397
The Rabin-Karp algorithm 401
Implementing the Rabin-Karp algorithm • 403
The Knuth-Morris-Pratt algorithm 406
The prefix function • 408

Understanding the KMP algorithm • 410

Implementing the KMP algorithm • 413

The Boyer-Moore algorithm **415**

Understanding the Boyer-Moore algorithm • 416

Bad character heuristic • 417

Good suffix heuristic • 420

Implementing the Boyer-Moore algorithm • 424

Summary **427**

Exercise **427**

Appendix: Answers to the Questions 429

Chapter 2: Introduction to Algorithm Design **429**

Chapter 3: Algorithm Design Techniques and Strategies **430**

Chapter 4: Linked Lists **432**

Chapter 5: Stacks and Queues **435**

Chapter 6: Trees **436**

Chapter 7: Heaps and Priority Queues **440**

Chapter 8: Hash Tables **442**

Chapter 9: Graphs and Algorithms **444**

Chapter 10: Searching **445**

Chapter 11: Sorting **447**

Chapter 12: Selection Algorithm **451**

Chapter 13: String Matching Algorithms **453**

Other Books You May Enjoy 461

Index 465