Chapter 1: Problem Solving

- What is an algorithm?

- Finding the maximum value in any list

- Counting key operations

- Models for predicting algorithm performance

- Finding the top two values in any list

- Tournament algorithm

- Time complexity and space complexity

- Chapter summary

- Challenges

Chapter 2: Algorithm Analysis

- Predicting performance with empirical models

- Faster multiplication

- Performance levels

- Asymptotic analysis

- Counting all operations

- Counting all bytes

- The door of destiny

- Binary array search

- As simple as π

- Killing two birds with one stone

- Overall

- Curve fitting and upper and lower bounds

- Chapter summary

- Challenges

Chapter 3: Hash Well, Live Well

- Relationship between values and keys

- Hash functions and hash values

- (Key, value) pair hash table structure

- Detecting and resolving conflicts with linear probing

- Separate linked lists

- Removing items from linked lists

- Evaluation

- Expanding hash tables

- Performance analysis of dynamic hash tables

- Perfect hash

- Iterative processing of (key, value) pairs

- Chapter summary

- Challenges

Chapter 4: Building Up Heaps

- Maximum binary heap

- Insertion (value, priority)

- Removing the highest priority value

- Representing a binary heap with an array

- Implementation of swim and sink

- Chapter summary

- Challenges

Chapter 5: Magical Sorting without Magic

- Sorting using exchanges

- Selection sort

- Analysis of quadratic sorting algorithms

- Performance analysis of insertion sort and selection sort

- Recursion and divide-and-conquer

- Merge sort

- Quick sort

- Heap sort

- Performance comparison of O(N log N) level algorithms

- Tim sort

- Chapter summary

- Challenges

Chapter 6: Binary Trees - Mastering Infinity

- Introduction

- Binary search tree

- Searching for content values in binary search trees

- Removing content values from binary search trees

- Traversing binary search trees

- Performance analysis of binary search trees

- Self-balancing binary search trees

- Performance analysis of self-balancing binary trees

- Implementing (key, value) symbol tables with binary trees

- Implementing priority queues with binary trees

- Chapter summary

- Challenges

Chapter 7: Graphs - All About Connections

- Efficiently storing useful information in graphs

- Solving mazes with depth-first search

- Applying different strategies for breadth-first search

- Directed graphs

- Weighted graphs

- Dijkstra's algorithm

- All pairs shortest paths

- Floyd-Warshall algorithm

- Chapter summary

- Challenges

Chapter 8: Conclusion

- Built-in data structures in Python

- Implementing stacks in Python

- Implementing queues in Python

- Implementing heaps and priority queues

- Future exploration