CS 231
Naomi Nishimura

Textbook coverage

1 Introduction

The primary source of information about the course material is the lectures, with some of the material also being available in written form, either as lecture slides or as extra resources available on the course website. The textbook recommended for the course should be seen as optional supplementary reference material, providing you with additional examples and exercises. When there is a discrepancy between terminology (such as the definition of *exhaustive search* versus *brute force*), pseudocode style, or other material found in both the textbook and course materials (including lecture slides and extra resources), the course materials have precedence.

Please read as much or as little of the textbook as you find helpful, using the lectures as a guide to the organization, importance, and presentation of topics. Keep in mind that the book does not cover all the material in the course; the textbook is a supplement to, not a replacement for, lectures and course material.

Because the textbook is not required, you are welcome to use older editions, with the caveat that the comments that appear here refer only to the current edition.

2 General comments on the Levitin book

The book "Introduction to the Design and Analysis of Algorithms" by Anany Levitin, third edition, covers most, but not all, of the course material, as well as related material that is not covered by the course. The biggest impediment to understanding the book is its use of data structures, which are covered in CS 234 but not in CS 231. Although I have omitted explicit coverage of data structures in the material recommended for reading, you will encounter the term *array*, which for our purposes can be viewed as a Python list.

The following sections are outside the scope of the course, with sections closer to course material being marked as "optional": 1.4 (with exceptions noted below), 2.5, 2.6, 2.7, 3.2 (optional), 3.3 (optional), 3.5, 4.1-4.3 (optional; this is a variant of divide-and-conquer), 4.5 (optional; this is a variant of divide-and-conquer), 5.2-5.5 (optional), 6, 7, 8.3, 9.4, 10, 11.4, 12.4.

3 Coverage by module

In the listings below, the phrase "P introduced" is used to indicate the first time problem P is introduced.

Module 1

Topics covered include:

- Types of problems (1.1-1.3)

- Graphs (pp. 28–31)
- Exhaustive search (3.4)
- TSP introduced (3.4)

Module 2

- Asymptotic notation (2.2 - skip pp. 56-58 on limits)
- Best, worst, and average cases (2.1)
- Analysis of pseudocode (2.3)

Module 3

- Greedy algorithms (9)
- Making change introduced (p. 287)
- Greedy algorithm for making change (p. 315)
- Selection sort (3.1; categorized a brute force by the book)
- Knapsack introduced (3.4 - general, not fractional knapsack)
- Single source cheapest paths introduced (p. 333)
- Greedy algorithm for single source cheapest paths (9.3)
- Minimum spanning tree introduced (p. 318)
- Greedy algorithm for Minimum spanning tree (9.1, 9.2 - skip pp. 327-331 on union-find)

Module 4

- Divide-and-conquer paradigm (5)
- Binary search (4.4)
- Recurrence relations (2.4)
- Mergesort (5.1)
- Iteration method (2.4 and pp. 480-481)
- Master theorem method (p. 171)

Module 5

- Dynamic programming paradigm (8)

- All pairs cheapest paths (8.4)
- Knapsack (8.2)

Module 6

- Information theory lower bounds (11.1)
- Decision trees (11.2)
- Adversary lower bounds (11.1)
- Reductions (11.1)
- NP (11.3)
- NP-completeness (11.3)

Module 7

- Backtracking (12.1)
- Branch-and-bound (12.2)
- Branch-and-bound for Knapsack (12.2)

Module 8

- Approximation algorithms (12.3)
- Metric TSP (12.3)
- Amortized algorithms (p. 49)

4 Python coverage

For coverage of Python-related material, please see the Python document.