Introduction to Algorithms

Syllabus

Class Meeting Times

Lectures: 1 hr / session; 2 sessions / week

Recitations: 1 hr / session; 2 sessions / week

Prerequisites

- *6.0001 Introduction to Computer Science and Programming in Python*: Basic experience programming in Python 3.
- *6.042J Mathematics for Computer Science*: Basic knowledge of discrete mathematics: set theory, relations and logic, combinatorics, proofs, recursion, number theory, graph theory, and probability.

We strongly caution against taking 6.006 before having fulfilled the listed prerequisites. We will evaluate your understanding of the prerequisite material via a short Problem Set 0 assignment. All students must submit this evaluation, regardless of prerequisite status. We will assign each submission a letter grade. If you receive a C or below on the assignment, you will need to meet with a staff member to review your performance before you will be allowed to take the class. We will not grade any other assignments from you until a good faith attempt of Problem Set 0 has been submitted. The grade for this assignment will NOT affect your final grade in the class, but turning it in is required for taking this class.

Course Description

This is an introductory course covering elementary data structures (dynamic arrays, heaps, balanced binary search trees, hash tables) and algorithmic approaches to solve classical problems (sorting, graph searching, dynamic programming). Introduction to mathematical modeling of computational problems, as well as common algorithms, algorithmic paradigms, and data structures used to solve these problems. Emphasizes the relationship between algorithms and programming, and introduces basic performance measures and analysis techniques for these problems.

Textbooks

Written course material will be distributed via notes from lectures and recitations. An additional useful reference is *Introduction to Algorithms* by Cormen, Leiserson, Rivest, and Stein (Third Edition, MIT Press) ISBN: 9780262033848, commonly known as *CLRS*, though this text is not required for the course.

Grading

Assignments	Percentages

Quiz 1	20%
Quiz 2	15%
Quiz 3	10%
Final Exam	35%
Problem Sets	18%
Recitation	2%

Calendar

Lectures occured on Tuesdays and Thursdays. Recitations took place on Wednesdays and Fridays. Optional problem sessions were held on Fridays.

Week	Lectures	Recitations	Problem Sessions	Key Dates
1	Lecture 1: Introduction Lecture 2: Data Structures	Recitation 1 Recitation 2	Problem Session 1	Problem Set 0 Due
2	Lecture 3: Sorting Lecture 4: Hashing	Recitation 3 Recitation 4	Problem Session 2	Problem Set 1 Due
3	Lecture 5: Linear Sorting	Recitation 5	Problem Session 3	Problem Set 2 Due
4	Lecture 6: Binary Trees, Part 1 Lecture 7: Binary Trees, Part 2: AVL	Recitation 6 Recitation 7	Problem Session 4	Problem Set 3 Due

5	Lecture 8: Binary Heaps Lecture 9: Breadth-First Search	Recitation 8 Recitation 9	Quiz 1 Review	Problem Set 4 Due Quiz 1 Review
6	Lecture 10: Depth-First Search Lecture 11: Weighted Shortest Paths	Recitation 10 Recitation 11	Problem Session 5	
7	Lecture 12: Bellman-Ford	Recitation 12	Problem Session 6	Problem Set 5 Due Quiz 1
8	Lecture 13: Dijkstra’s Algorithm Lecture 14: Johnson’s Algorithm	Recitation 13 Recitation 14	Problem Session 7	Problem Set 6 Due Quiz 2 Review
9	Lecture 15: Dynamic Programming, Part 1: Recursive Algorithms Lecture 16: Dynamic Programming, Part 2: Subproblems	Recitation 15 Recitation 16	Problem Session 8	
10	Lecture 17: Dynamic Programming, Part 3: APSP, Parens, Piano	Recitation 17	No problem sessions	Problem Set 7 Due Quiz 2
11	Lecture 18: Dynamic Programming, Part 4: Pseudopolynomials Lecture 19: Complexity	Recitation 18 Recitation 19	Problem Session 9	Problem Set 8 Due Quiz 3 Review

12	Lecture 20: Course Review	Recitation 20	No problem sessions	Quiz 3
13	Lecture 21: Algorithms—Next Steps	No recitations	No problem sessions	
14	Final Exam			