**Course Meeting Times**

Lectures: 2 sessions / week, 1 hour / session

Recitations: 2 sessions / week, 1 hour / session

**Lecture Topics**

Unit 1: Introduction

1 Algorithmic thinking, peak finding Problem set 1 out

2 Models of computation, Python cost model, document distance

Unit 2: Sorting and Trees

3 Insertion sort, merge sort

4 Heaps and heap sort

5 Binary search trees, BST sort

6 AVL trees, AVL sort Problem set 2 due

7 Counting sort, radix sort, lower bounds for sorting and searching

8 Hashing with chaining

9 Table doubling, Karp-Rabin

10 Open addressing, cryptographic hashing

11 Integer arithmetic, Karatsuba multiplication

12 Square roots, Newton’s method

Unit 5: Graphs

13 Breadth-first search (BFS)

14 Depth-first search (DFS), topological sorting

Unit 6: Shortest Paths

15 Single-source shortest paths problem

16 Dijkstra

17 Bellman-Ford

18 Speeding up Dijkstra

19 Memoization, subproblems, guessing, bottom-up; Fibonacci, shortest paths

20 Parent pointers; text justification, perfect-information blackjack

21 String subproblems, psuedopolynomial time; parenthesization, edit distance, knapsack

22 Two kinds of guessing; piano/guitar fingering, Tetris training, Super Mario Bros.

23 Computational complexity

24 Algorithms research topics

**Course Description**

This course provides an introduction to mathematical modeling of computational problems. It covers the common algorithms, algorithmic paradigms, and data structures used to solve these problems. The course emphasizes the relationship between algorithms and programming, and introduces basic performance measures and analysis techniques for these problems.

**Prerequisites**

A firm grasp of Python and a solid background in discrete mathematics are necessary prerequisites to this course. You are expected to have mastered the material presented in [*6.01 Introduction to EECS I*](https://ocw.mit.edu/courses/6-01sc-introduction-to-electrical-engineering-and-computer-science-i-spring-2011) and [*6.042J Mathematics for Computer Science*](https://ocw.mit.edu/courses/6-042j-mathematics-for-computer-science-fall-2010).

If you have not taken and been successful in each of these subjects, please speak with a TA or professor before enrolling. We do allow students who have equivalent, other experience with the material described above to enroll, but with the firm understanding that mastery of this material is assumed and that course staff will not feel obligated to cover it or to help students who are struggling with it.

6.006 is a 12-unit (4-0-8) subject and serves as a Foundational Computer Science subject under the new curriculum. It is a direct prerequisite for *6.046 Design and Analysis of Algorithms,* the theory header.

**Textbooks**

**Required**

[Buy at MIT Press](https://mitpress.mit.edu/9780262033848) Cormen, Thomas, Charles Leiserson, Ronald Rivest, and Clifford Stein. [*Introduction to Algorithms*](https://mitpress.mit.edu/9780262033848). 3rd ed. MIT Press, 2009. ISBN: 9780262033848.

For the student who finds books helpful, we also suggest:

Miller, Bradley, and David Ranum. *Problem Solving with Algorithms and Data Structures Using Python*. 2nd ed. Franklin, Beedle & Associates, 2011. ISBN: 9781590282571.

**Software**

[6.006 programming environment setup](https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-fall-2011/pages/syllabus/software/)

**Lectures and Recitations**

One-hour lectures are held twice a week. You are responsible for material presented in lectures, including oral comments made by the lecturer (or other information that may not be present in the notes).

One-hour recitations are held twice a week, one day after the lectures. You are responsible for the material presented in recitation, which may include new material not presented in lectures. Recitation attendance has been well-correlated with quiz performance in past semesters. Recitations also give you a more intimate opportunity to ask questions of and to interact with the course staff. Your recitation instructor is responsible for determining your final grade.