Course Information

Instructors

Zachary Abel	ZA	zabel@mit.edu	38-652
Erik Demaine	ED	edemaine@mit.edu	32-G680
Jason Ku	JK	jasonku@mit.edu	38-683

Teaching Assistants

Michael Coulombe	MC	mcoulomb@mit.edu
Ivan Ferreira	IF	ivanaf@mit.edu
Courtney Guo	CG	ckguo@mit.edu
Justine Jang	JJ	justinej@mit.edu
Alexander Katz	AK	alexkatz@mit.edu
Vivek Miglani	VM	vivekm@mit.edu
Preksha Naik	PR	prekshan@mit.edu
Patricio Noyola	PN	pnoyola@mit.edu
Stephanie Ren	SR	stefren@mit.edu
Alap Sahoo	AS	asahoo@mit.edu
Rose Wang	RW	rewang@mit.edu

Questions?

6.006-questions@mit.edu

Websites

Stellar	Announcements, calendar, grades, and PDF course content. https://tinyurl.com/6-006fa18
Piazza	**All** discussion related to course material. https://piazza.com/class/jlmfeigsnrwjw
Gradescope	LaTeX problem set submissions, quiz grading, and regrades. https://gradescope.com/courses/24353
Code Checker	Auto-graded code problem set submissions. https://alg.mit.edu

Content

6.006 is an introductory course in algorithms and data structures. It covers elementary data structures (dynamic arrays, heaps, balanced binary search trees, hash tables) and algorithmic paradigms (brute force, divide and conquer, greedy, dynamic programming). The material is applied to classical motivating problems including searching, sorting, and finding shortest paths in a graph. 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), commonly known as **CLRS**, though this text is not required for the course.

Prerequisites

6.0001	Basic experience programming in Python version 3.
6.042	Basic knowledge of discrete mathematics, specifically combinatorics and set theory, logic and proofs, recursion and invariants, graph theory, and probability.

We **strongly** caution against taking 6.006 before having fulfilled these prerequisites; however, we are leaving it up to each individual student to decide whether to take the class without them. If you are missing a prerequisite, please be sure to study this material on your own as you will be responsible for material covered in the perquisite courses.

Instruction

Lectures will be held in Room 26-100 from 11:05 A.M. to 11:55 A.M. on Tuesdays and Thursdays. One-hour **Recitations** will be held weekly on Wednesdays and Fridays, beginning on Friday, September 7th. Recitations supplement the material presented in lecture in a more interactive setting. You are responsible for material presented during both lecture and recitation. Initial recitation assignments have been provided by the registrar, though you may reassign yourself to any recitation section with available space via the Stellar website. Please finalize your recitation section choice by Wednesday, September 19th.

R01 @10 34-304 VM	R02 @10 34-302 JJ	R03 @10 26-310 RW	R04 @11 34-304 AS	R05 @11 34-302 SR	R06 @11 36-155 PN	R07 @12 34-301 CG
R08 @12 34-304 AS	R09 @1 34-304 CG	R10 @1 35-310 AK	R11 @2 35-310 MC	R12 @2 26-204 IF	R13 @3 26-204 IF	R14 @4 26-204 VM

Office Hours

Teaching Assistants will hold office hours from 7-10 P.M. on Sundays, Mondays, Tuesdays, Wednesdays, and Thursdays, each day in rooms 26-302 and 26-322, beginning on Sunday, Sep. 9. Instructors will hold individual office hours by appointment.

Grading Policy

Your grade will be based on the following assignments: 10 problem sets, 2 quizzes, and final exam.

	Weight	Date	Time	Location(s)
Quiz 1	20%	Tuesday, October 16, 2018	7:30–9:30 P.M.	26-100 & 32-123
Quiz 2	20%	Thursday, November 15, 2018	7:30–9:30 P.M.	26-100 & 32-123
Final Exam	34%	Thursday, December 20, 2018	1:30–4:30 P.M.	Johnson Track
Problem Sets	25%	10 Problem Sets, 2.5% each		
Recitation	1%	Graded by your recitation instructor		

Problem Sets

PS	Release	Due	Topic
01	R 9/06	R 9/13	Asymptotics, Recurrences
02	R 9/13	R 9/20	Sorting / Dynamic Arrays
03	R 9/20	R 9/27	Trees / Heaps
04	R 9/27	R 10/04	BST / AVL
05	R 10/04	R 10/11	Integer Sorting / Hashing
06	R 10/18	R 10/25	Graph Traversal
07	R 10/25	R 11/01	Bellman-Ford / Dijkstra
08	R 11/01	U 11/11	Shortest Paths / Dynamic Programming
09	R 11/16	R 11/29	More Dynamic Programming
10	R 11/29	R 12/06	Everything Review

Each problem set will contain a theory portion and a coding portion. Each theory portion must be uploaded to Gradescope as a PDF file compiled from a provided LaTeX template. Each coding portion will be administered and automatically graded via our Code Checking website, and must be completed using Python version 3. Problem set submissions are **due by 11 P.M.** on the posted due date.

Late submissions will be accepted up until 48 hours after the due date, also at 11 P.M.. Solutions will be posted shortly after the late submission window closes. We will not penalize your two highest scoring late submissions, but we will penalize any additional late submissions by 50%. In exceptional circumstances, problem set deadlines may be individually extended without penalty at the emailed request of an Institute Dean, to `6.006-questions@mit.edu`.

Collaboration

The goal of the problem sets is for you to practice applying the course material. In this class, you are **encouraged** to collaborate on problem sets. Students who work together on problem sets generally do better on exams than students who work alone, but you will learn the material best if you **work on the problems FIRST on your own**. Some forms of collaboration are **not allowed**; some examples are listed below. Violating the collaboration policy to increase your score on a problem set is likely to lower your score on an exam, which carries significantly more weight. A violation may also lead to academic action and/or a significant penalty on your grade.

- Identify your collaborators in the space provided in each LaTeX template.
- Write code and theory problem solutions **by yourself** in your own words.
- Do **NOT** directly copy the work of others.
- Do **NOT** look at written solutions or code by other students code before submitting your own solution. You may look at another student's code on their screen, only to help them debug, and only after you have submitted your own solution.
- Do **NOT** let other students see your written solutions.
- Do **NOT** send other students your code.
- You may ask TAs to help you debug your code during office hours.

Exams

There will be no official lecture on quiz days. A review session will be given during the recitation preceding each quiz. Quizzes and the Final Exam will be closed book, but you will be allowed to bring and use some pre-prepared **double-sided notes**: one page for Quiz 1, two pages for Quiz 2, and three pages for the Final Exam.

Attendance at the quizzes and the Final Exam is mandatory and may not be excused. A quiz may be rescheduled at the emailed request of an Institute Dean, to `6.006-questions@mit.edu`. Course-wide makeup quizzes will be given within a day or two of the quiz, or for the Final Exam at a time scheduled by the registrar.

Regrade Requests

If you feel that any assignment has been graded incorrectly, you may submit a regrade request to the relevant assignment in Gradescope within one week after the assignment's grade has been released. For any regrade request, we reserve the right to regrade the entire assignment, and your grade may be adjusted up or down as a result of the regrade.

Syllabus

	Date	Lec	Topic		Date	Rec	Topic
R	9/06	L01	Algorithms and Computation	F	9/07	R01	Asymptotics
T	9/11	L02	Design Paradigms	W	9/12	R02	Recurrences / Master Theorem
R	9/13	L03	Insertion Sort / Merge Sort	F	9/14	R03	In-Place Merge Sort
T	9/18	L04	Data Structures / Dynamic Arrays	W	9/19	R04	Python Lists
R	9/20	L05	Heaps / Priority Queues	F	9/21		Career Week Holiday
T	9/25	L06	Binary Search Trees	W	9/26	R05	Heaps & Trees in Python
R	9/27	L07	Height Balanced AVL Trees	F	9/28	R06	AVL Tree in Python
T	10/02	L08	Direct Access & Hashing	W	10/03	R07	Dicts and Sets
R	10/04	L09	Linear Sorting	F	10/05	R08	Integer Sorts in Python
T	10/09		Columbus Day	W	10/10	R09	Data Structures Review
R	10/11	L10	Breadth-First Search	F	10/12	R10	Quiz 1 Review
T	10/16		**Quiz 1: L01 – L09**	W	10/17		
R	10/18	L11	Depth-First Search	F	10/19	R11	Graph Traversal in Python
T	10/23	L12	Weighted Shortest Paths	W	10/24	R12	Relaxation
R	10/25	L13	Bellman-Ford	F	10/26	R13	Bellman-Ford in Python
T	10/30	L14	Dijkstra	W	10/31	R14	Dijkstra in Python
R	11/01	L15	All-Pairs Shortest Paths	F	11/02	R15	Johnson's Algorithm & Review
T	11/06	L16	Dynamic Programming Intro	W	11/07	R16	Dynamic Programming in Python
R	11/08	L17	Guessing Subproblems	F	11/09	R17	Dynamic Programming Examples
T	11/13	L18	Subproblem Expansion	W	11/14	R18	Quiz 2 Review
T	11/15		**Quiz 2: L01 – L17**	F	11/16		
T	11/20	L19	Fun with Algorithms	W	11/21		
R	11/22		Thanksgiving	F	11/23		
T	11/27	L20	Subset Sum & Pseudo-polynomial	W	11/28	R20	Partition & 0-1 Knapsack
R	11/29	L21	P, NP, Hardness, Completeness	F	11/30	R21	Complexity
T	12/04	L22	Specialized Algorithms	W	12/05	R22	Advanced Examples
R	12/06	L23	Course Review	F	12/07	R23	Final Exam Review
T	12/11	L24	Algorithms Research	W	12/12	R24	Final Exam Review
R	12/13			F	12/14		
			Final: L01 – L24				