Course Information

This handout describes basic course information and policies. Please read it carefully as it contains important information that will be useful throughout the semester. The main items to pay attention to **NOW** are:

1. Please register for an alg.csail account by **5:00 P.M. today** (Thursday, September 8) at `https://alg.csail.mit.edu/6.006/` to receive a recitation assignment.
2. Recitations will start on Friday, September 9th. **Please ignore your recitation assignment from the registrar.** You will be assigned a recitation through alg.csail.
3. Please note the date of the two evening quizzes on the calendar and plan trips accordingly. Also note that this class will have a final exam, whose date will fall during finals period.
4. Please note the collaboration policy for homework.
5. Please note the grading policy.

1 Staff

To contact all the course staff, please email `6.006-staff@mit.edu`

To contact instructors only, please email `6.006-faculty@mit.edu`

Instructors:	Erik Demaine	32-G680	`edemaine@mit.edu`
	Debayan Gupta	32-G883	`debayan@mit.edu`
	Ronitt Rubinfeld	32-G698	`ronitt@csail.mit.edu`
Teaching Assistants:	Anton Anastasov		`aanastasov@alum.mit.edu`
	Themistoklis Gouleakis		`tgoule@mit.edu`
	Adam Hesterberg		`achester@mit.edu`
	Gregory Hui		`ghui@mit.edu`
	Atalay Ileri		`atalay@mit.edu`
	Skanda Koppula		`skoppula@mit.edu`
	Akshay Ravikumar		`akshayr@mit.edu`
	Ludwig Schmidt		`ludwigs@mit.edu`
	Yonadav Shavit		`yonadav@mit.edu`
	Aradhana Sinha		`aradhana@mit.edu`
	Ali Vakilian		`vakilian@mit.edu`
	Ray Hua Wu		`dzaefn@mit.edu`
	Parker Zhao		`pzarker@mit.edu`
	Daniel Zuo		`dzuo@mit.edu`

Websites: For announcements, information and handouts, tutorial material, lecture notes, calendar, and office hours
`https://learning-modules.mit.edu/class/index.html?uuid=/course/6/fa16/6.006`

For recitation assignments and problem sets
`https://alg.csail.mit.edu/6.006/`

For questions about homework and materials
`https://piazza.com/mit/fall2016/6006/`

2 Prerequisites

A strong understanding of programming in Python and a solid background in discrete mathematics are necessary prerequisites to this course. You are expected to have taken 6.01 *Introduction to EECS I* (or coreq 6.009 *Fundamentals of Programming*) and 6.042J/18.062J *Mathematics for Computer Science*, and received a grade of C or higher in both classes.

We will use Python (version 3) programming extensively, particularly in the homework. Students having completed 6.01 should have sufficient experience with Python to be ready for 6.006. We will also make use of basic discrete mathematics, combinatorics, graph theory, basic logic and proofs, and invariants. These are taught in 6.042 and not in other common alternatives.

We are leaving it up to each individual student to decide whether to take the class without one or both of these prerequisites, but caution that the semester could be much more difficult for such students. For the case of 6.042, some students have other experience in their background that covers this material, and others manage to learn it on their own in time for use in 6.006. Seeking online materials related to 6.042, such as lecture notes or other reading, might be helpful in this regard. If you want to speak in detail about your case, it would be good to discuss it with your TA at your first recitation.

3 Course 6 requirements

6.006 serves as a Foundational Computer Science subject. It is a prerequisite for 6.046, which serves as a Computer Science theory header.

4 Textbook

The primary written reference for the course is the Third Edition of the textbook *Introduction to Algorithms* by Cormen, Leiserson, Rivest, and Stein (MIT Press).

5 Lectures

Lectures will be held in Room 32-123 from 11:05 A.M. to 11:55 A.M. ET on Tuesdays and Thursdays. You are responsible for material presented in lectures, including oral comments made by the lecturers.

6 Recitations

One-hour recitations will be held weekly on Wednesdays and Fridays, beginning on Friday, September 9th (tomorrow). **Wednesday and Friday recitations have different content, and you are responsible for material presented both on Wednesdays and on Fridays.** In the past, attendance in recitation has been well correlated with exam performance. Recitations also give you an opportunity to ask questions and interact with the teaching staff in a more personal environment.

We do **NOT** use the recitation assignments made by the scheduling office. During the first meeting of the class, the lecturer will tell you how to register on the course webpage. The course staff will assign recitations based on the information you provide on the course website `https://alg.csail.mit.edu/6.006/` (by running an appropriate algorithm that minimizes the number of conflicts). **Please complete the online form by 5:00 P.M. today (Thursday, September 8)**. Recitation assignments will be posted shortly thereafter to the course alg.csail website. If you cannot make any recitation time, you should not take the class.

7 Problem sets

Five problem sets will be assigned during the semester.

Each problem set will contain a theory portion and a coding portion. We believe that for many students it is advisable to complete the coding portion before the theory, to help develop the concepts that are built upon in the theory portion. We will be providing encouragement throughout the semester to complete and submit the coding portion in advance of the overall problem set deadline. The coding portion will be done in Python (version 3).

All submissions (both code and written components) must be uploaded to `https://alg.csail.mit.edu/6.006/` **(NOT STELLAR / LEARNING MODULES)**. You can revise your answer repeatedly until the deadline; only the *final* submission will be graded.

- Each student is allotted five "grace days" of 24 hours each to spread across the five assignments. Grace days must be used in increments of a day, and no more than two grace days may be used on any assignment (and *only one grace day on the last assignment*, to follow MIT regulations). **Students should email their recitation instructor in advance to notify them that they will be exercising their grace day option**.
- **We expect students to use grace days for both planned and unforeseen circumstances (illness, required travel, conflicts and overload, etc.)**. Beyond the use of grace days, late homework will not be accepted without a note from the Dean's Office and, even then, only in truly extreme circumstances.
- **Problem set solutions must be typeset in LaTeX, not handwritten**. Each problem set will provide a LaTeX template to start from. You should compile your LaTeX into PDF (using `pdflatex`) and upload the PDF file to the course website by the date and time specified. If hand-drawn diagrams are useful for explaining solutions, please refer to the diagrams in your LaTeX submission, scan them, and include them in your submission. **Handwritten solutions will not be accepted.** Knowing LaTeX is a useful life skill and algorithms is a context where it is especially helpful.
- Be sure to fill in the "Collaborators" section of each problem. If you solved the problem alone, write "none".

Guide to writing up homework

You should be as clear and precise as possible in your write-up of solutions. Understandability of your answer is as desirable as correctness, because communication of technical material is an

important skill.

A simple, direct analysis is worth more points than a convoluted one, both because it is simpler and less prone to error, and because it is easier to read and understand. Sloppy answers will receive fewer points, even if they are correct, so make sure that your solutions are clear and concise.

You will often be called upon to "give an algorithm" to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of your essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudocode.
2. At least one worked example or diagram to show more precisely how your algorithm works.
3. A proof (or indication) of the correctness of the algorithm.
4. An analysis of the asymptotic running time behavior of the algorithm.

Remember, your goal is to communicate. Graders will be instructed to take off points for convoluted and obtuse descriptions. We will organize problem set labs (essentially pset-specific office hours) before each problem set; schedule details will be on the course website.

8 Exams

There will be two quizzes on the dates and time windows indicated:

- Tuesday, October 18th, 2016, 7:30–9:30 P.M.
- Thursday, November 17th, 2016, 7:30–9:30 P.M.

There will be no lecture on those two days, and the recitation on the previous day will be a review session. There will also be a final exam during finals week. Each of the quizzes and the final exam will be closed book. However, **you will be allowed to bring and use one double-sided, letter-sized piece of paper with your own notes for the first quiz, two for the second quiz, and three for the final**. These should not be necessary but could be helpful.

9 Grading policy

The final grade will be determined based on 5 problem sets, 2 quizzes, and a final.

- The five problem sets will together be worth 30% of the overall grade.
- The quizzes will be worth 20% each (40% total).
- The final exam will be worth 30%.

Attendance at the quizzes and the final is mandatory. Legitimate conflicts can be discussed with the teaching staff but must be due to extenuating circumstances and discussed in advance. If a student misses either quiz or the final exam due to an emergency, a note from the Dean's Office will be required justifying his or her absence before a makeup will even be considered. If a makeup request is granted, the student will be expected to attend a pre-scheduled, course-wide makeup that will take place a day or two after the quiz, and a time scheduled by the Registrar for the final exam.

Regrade requests

Any student who feels that a problem set, quiz, or final exam was not graded properly may submit a regrade request. The request must be made in writing (email) to their recitation instructor within one week of the graded assignment being returned to the student. The request should include a detailed explanation of why they believe that a regrade is warranted. The course staff reserves the right to regrade the entire assignment.

Grading of Code

Code will be graded by running a series of unit tests and by examining the algorithm and code. For full credit code must satisfy the following criteria:

Correctness and Running Time Your code must pass the unit tests with the correct answers and within the allotted running time. You will be given all of the unit tests in advance so you are able to test your code on your own before submitting it.

Algorithm and Code Your code must be well documented with comments describing the algorithm you used. Your code must be readable so the TAs are able to verify that your code does what it is supposed to do. Your algorithm should be asymptotically efficient.

10 Collaboration policy

The goal of homework is to give you practice in mastering the course material. Consequently, you are encouraged to collaborate on problem sets. In fact, students who form study groups generally do better on exams than do students who work alone. If you do work in a study group, however, you owe it to yourself and your group to be prepared for your study group meeting. Specifically, you should spend at least 30–45 minutes trying to solve each problem beforehand. If your group is unable to solve a problem, talk to other groups, visit office hours, or ask your recitation instructor.

You must write up each problem solution by yourself without assistance, even if you collaborate with others to solve the problem. You are asked on problem sets to identify your collaborators. If you did not work with anyone, you should write "Collaborators: none." You should always write whatever solution you come up with in your own words.

Code you submit must also be written by yourself. A suite of algorithms will be run to detect plagiarism in code. No other 6.006 student may use your solutions; this includes your writing, code, tests, documentation, etc. Similarly, you may not use or access anyone else's solution.

We have, unfortunately, encountered gross violations of the collaboration policy in past 6.006 terms and have had to report several cases to the Committee on Discipline, with drastic consequences for some of the students involved.

11 Advice and resources for effective learning

Because of the conceptual nature of the material, just attending lectures and recitations and doing the homework, are unlikely to be sufficient for learning all the concepts. **Setting aside time to**

do the reading and to study your notes from lecture and recitation is generally necessary to truly learn and internalize the material, and to be able to apply it in new ways later in the course as well as for the rest of your life.

Homework is essential for learning the material. Rather than thinking of problem sets as just a requirement, recognize it as an excellent means for learning the material, and for building upon it. Spread out the time you have to work the problems. Many people learn best by reading the problems long before they are due, and working on them over the course of one or two weeks; they find that their minds make progress working the problems in the background or during downtime throughout the day. Few people do their best learning the night before an assignment is due. Work with others if that is helpful, but with the goal of learning first and solving the problems second. **It is worth reading the posted homework solutions, even if you received full credit**. Often the clarity of explanation or details of implementation are different from the way you were thinking about things in ways that can improve your learning.

Recitation is an integral part of this class and extends the material presented in lecture. Recitation allows you to ask questions and discuss the material in a more intimate environment. Preparation for recitation can pay off in big dividends. Office hours are an essential means of filling in any gaps in your understanding and overcoming difficulties you may be having with the problem sets. There is no need to attend only the office hours of your own recitation instructor; attend whatever office hours are convenient for you. Preparing for office hours, as with recitation, can make everything more productive.

Don't hesitate to ask for help. This class is largely conceptual, and the concepts tend to build on one another. **If you are having trouble understanding the material, it is important to catch up rather than risk falling further behind**. We can help.

Extra help may be obtained from the following two resources. The MIT Department of Electrical Engineering and Computer Science provides one-on-one peer assistance in many basic undergraduate Course VI classes. Tutoring is free. New this year, 6.006 staff can send recommendations to HKN on behalf of students who might benefit the most from this service; such students will be prioritized in tutor assignment. More information is available on the HKN web page: `https://hkn.scripts.mit.edu/tutoring/`

Tutoring is also available from the Talented Scholars Resource Room (TSR^2) sponsored by the Office of Minority Education. The tutors are undergraduate and graduate students. For further information, go to `http://web.mit.edu/tsr/`

This class has great material, so HAVE FUN!