Design and Analysis of Algorithms, 6.046/18.410, Spring 2014

Massachusetts Institute of Technology

Lecturers: Manolis Kellis, Michael Kapralov, Ran “RON” Libeskind-Hadas
Teaching Assistants: Shalev Ben David, Dawsen Hwang, Vlad Kontsevoi, Hayden Metsky, Ofir Nachum, Kang Zhang

Lecture Time and Place: Tuesdays and Thursdays, 9:30-11 AM, 34-101
Course Website: https://stellar.mit.edu/S/course/6/sp14/6.046
Course Staff Email: 6046-staff@mit.edu

What is this Course About?

Computer scientists are frequently confronted with new problems that require new algorithms. This course endeavors to teach you general and powerful techniques that will allow you to design, analyze, and prove the correctness of your own algorithms. In addition, you’ll learn about some beautiful existing algorithms.

By the time you’ve completed this course, you should be able to do the following:

- Design algorithms using major paradigms including the greedy method, divide-and-conquer, and dynamic programming
- Choose the appropriate data structures to support the development of efficient algorithms
- Analyze data structures and algorithms using amortization methods
- Analyze the asymptotic worst-case running time of your algorithm
- Prove the correctness of your algorithm using induction, loop invariants, and other methods
- Analyze the average-case running time of probabilistic algorithms
- Explain basic properties of randomized algorithms and methods for analyzing them
- Explain major graph algorithms and know how to apply and modify them to solve an array of problems that arise in numerous applications
- Understand how to formulate certain optimization problems as linear programs and the computational complexity of various linear programming approaches
- Understand the notion and importance of NP-completeness and be able to prove that a problem is NP-complete
- Be familiar with some “advanced” ideas including approximation algorithms and schemes, online algorithms, computational geometry, among others
Lectures

Please come to lecture! We’re eager to see you there and we hope that you’ll be eager to attend. You’re responsible, of course, for all material covered in lecture, some of which is neither in the book nor in the lecture notes.

Recitations

You’ve already been assigned to your recitation time by the registrar. Recitations will be held on Fridays by the TAs and will cover new material in addition to reviewing concepts from the week’s lectures. You’re responsible for the material covered in recitation and your final grade will be assigned, in part, by your recitation TA.

Prerequisites

This course is the header course for the MIT/EECS Engineering Concentration of Theory of Computation. You’re expected to have already taken…

- 6.006 Introduction to Algorithms and
- either 6.042J/18.062J Mathematics for Computer Science or 18.310 Principles of Applied Mathematics

…and received grades of C or better in those courses.

Problem Sets

In our experience, the best way to master the material, do well on the exams, and generally succeed in this course is to have worked through the exercises and problem sets. The problems are challenging and thought-provoking, but we also hope that you’ll find them useful and rewarding.

How many? There will be five problem sets assigned during the semester. Each set will comprise problems to be submitted and exercises that we strongly recommend you work on but are not to be submitted. The exercises will help you master the material and prepare you to solve the problems, prepare for the exams, and generally develop a stronger mastery of the material.

When are problems sets due? The due dates will appear on the top of each problem set. They will be due at 11:59 PM on the specified due date and should be submitted as pdf’s through the Stellar website.
**What's the late homework policy?** Late homework will not be accepted. If there are extenuating circumstances, you should make prior arrangements with your recitation instructor as soon as you know of your conflict and *at least one week in advance*. The more lead time you provide, the more likely the course staff will be able to accommodate your request. Authorization from the Dean's Office will be required if prior arrangements have not been made.

**How should problem sets be formatted?** While you're strongly encouraged to use LaTeX, you're welcome to use a different typesetting system or even scan in hand-written solutions. All submissions must be in pdf format and must be clear and legible. The graders can only give you credit for what they can read. Each problem must be submitted separately, since problems will generally be graded by separate graders. Mark the top of each sheet with the following: (1) your name, (2) the question number, and (3) the names of any people you worked with on the problem (see “Collaboration Policy” below), or “Collaborators: none” if you solved the problem completely alone.

**How are the problems graded?** Problems will be graded on a 5 point scale.

**Will we get sample solutions?** Sample solutions to the problems will be available shortly after the problem set is due. We do not provide sample solutions to the exercises, so we encourage you to compare your answers with those of your classmates and come to office hours if you’d like to discuss the exercises.

**Guide to Writing Up Homework**

Good technical writing is an invaluable skill that you’ll have a chance to practice and refine in this course. A brilliant solution that is difficult for a reader to understand will go unrecognized (e.g., by your boss, your peers, journals). Thus, in this course, a substantial fraction of the points will be allocated to precision and clarity of exposition. Handout 2 provides a sample problem and solution.

Here are some general guidelines for writing good solutions:

- Simple and elegant solutions are better than complicated and convoluted ones because they are easier to understand and generally easier to prove correct.
- Solutions should be written in clear, typo-free, and grammatically correct English.
- When describing an algorithm, use clear English. If you feel that pseudo-code would be helpful, you’re welcome to add that as well. Generally, pseudo-code alone is hard to read and doesn’t provide the reader with enough context or intuition to discern the big ideas, so use pseudo-code to support your English description rather than to
replace it. Finally, if you use pseudo-code, use very high-level syntax rather than real code (e.g., C++ and Java are not pseudo-code - they are encumbered by syntactic details that hinder understanding the main ideas in an algorithm).

- When asked to develop an efficient algorithm, faster algorithms will receive more credit than slower algorithms.
- Before submitting, re-read your solutions to check for errors large and small. Ask yourself “if someone handed this to me to read, would I be able to understand it without too much effort?”

**Exams**

This course will have two quizzes and one final exam. The tentative time for the quizzes is as follows:

- **Quiz 1**: Thursday, March 13, 7-9 PM, Room 32-123
- **Quiz 2**: Take-home exam given out in class on Thursday, April 17 and submitted online by 11:59 PM on Wednesday, April 23

There will be no scheduled make-up quizzes. If you have a conflict, let us know as soon as possible. The sooner you tell us, the more likely we can accommodate your request.

The dates for all final exams will be published by the MIT Registrar’s office on February 20.

**Grading policy**

The final grade will be based on five problem sets, one in-class quiz, one take-home quiz, and the final exam.

The grading breakdown is as follows:

- Problem sets: 25%
- In-class quiz: 20%
- Take-home quiz: 25%
- Final exam: 30%

**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. 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 45 minutes trying to solve each problem beforehand. If your group is unable to solve a problem, talk to other groups or ask your recitation instructor.

Appropriate collaboration means talking and sketching ideas on a board. The actual write-up of each solution must be done independently. For each solution that you submit, you should identify your collaborators. If you did not work with anyone, you should write “Collaborators: none.” Sharing written solutions or looking for written solutions in books, online, solution sets from previous years, or in other media is a violation of academic integrity. For more on MIT’s academic integrity policies and advice about working under pressure, please see http://integrity.mit.edu.

No collaboration whatsoever is permitted on quizzes or exams.

If you have any questions about the collaboration policy, or if you feel that you may have violated the policy, please talk to a member of the course staff. Although the course staff is obligated to deal with cheating appropriately, transgressions that we discover must be dealt with more severely than those that are self-reported.

Course website

The course website contains links to electronic copies of handouts, corrections made to the course materials, and special announcements. You should visit this site regularly to be aware of any changes in the course schedule, updates to your instructors’ office hours, etc.

Extra Help

You’re encouraged to post questions on the course Piazza site at:

https://piazza.com/mit/spring2014/6046/home

Course office hours will be set based on polling the class. These office hours will be announced in class and posted on the course webpage. You may attend the office hours of any TA (not just your own).

Further help may be obtained through tutoring services. The MIT Department of Electrical Engineering and Computer Science provides one-on-one peer assistance in many basic undergraduate Course VI classes. During the first nine weeks of the term, you may request a tutor who will meet with you for a few hours a week to aid in your understanding of course material. You and your tutor arrange the hours that you meet, for your mutual convenience. This is a free service. More information is available on the HKN web page: https://hkn.mit.edu/tutoring
Tutoring is also available from the Tutorial Services Room (TSR) sponsored by the Office of Minority Education. The tutors are undergraduate and graduate students, and all tutoring sessions take place in the TSR (Room 12-124) or the nearby classrooms. More information is available at: http://web.mit.edu/tsr/www