Syllabus

Fall 2016

Time and Location: Tuesdays and Thursdays, 2:30 – 3:50 PM in Barus and Holley room 168.

Collaboration Hours: Up to date collaboration hours for our entire course staff are posted on our website, `cs.brown.edu/courses/cs157/hours.shtml`.

Textbooks: The first textbook is a convenient and inexpensive paperback, which concisely introduces many of the topics we cover; we recommend buying it or obtaining a digital copy. The second (optional) textbook is encyclopedic, and a useful reference for more topics.

- *Algorithms* by Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, McGraw Hill 2008, ISBN 978-0-07-352340-8.
- *Introduction to Algorithms, 3rd edition* by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, MIT Press 2009, ISBN 978-0-262-03384-8.

Course Staff

Professor	Office	Email (@cs.brown.edu)
Prof. Paul Valiant	CIT 379	`pvaliant`

Head Teaching Assistants	@cs.brown.edu
Jemma Issroff	`jissroff`
Rajesh Jayaram	`rjayara1`
Samier Merchant	`samercha`

Graduate Teaching Assistants	@cs.brown.edu
Jasper Lee	`clee10`

Undergraduate Teaching Assistants	@cs.brown.edu
Athyuttam Eleti	`aeleti`
Brad Bentz	`bbentz`
Christopher Shum	`cshum`
Ian Boros	`iboros`
Joseph Engelman	`joengelm`
Kevin Cole	`kbcole`
Louisa Conwill	`lconwill`
Miles Holland	`mh91`
Martin Zhu	`mjzhu`
Michael Mueller	`mmuelle1`
Nathan Malimban	`nmalimba`
Philip Strauss	`pgstraus`
Ruby Goldberg	`rmg1`
Uthsav Chitra	`uchitra`
Xiangyu (Tommy) Yao	`xy24`
Ze (Bill) Li	`zl47`

Introduction

Welcome to CS157, Design and Analysis of Algorithms.

The course satisfies the theory requirement for the Sc.B. track in Computer Science.

Prerequisites

In order to take CS157, you should have taken (CS16 or CS18 or CS19) and (CS22 or CS145) or their equivalents. If you are a graduate student or a non-CS concentrator, these prerequisites are intended to cover three things: a semester of programming (in any language); a semester of very basic exposure to algorithmic concepts (for example, Dijkstra's algorithm, sorting algorithms, and big-O notation); and a semester that introduces proofs in a computer science or discrete math context (for example, proof by induction).

This course will emphasize both theoretical and practical aspects of algorithms. For the theoretical side, we require "mathematical maturity," which means either previous exposure to fundamentals like matrices and complex numbers, or a willingness to get caught up quickly. In addition, we will enforce a rigorous and formal proof style throughout the course to help us gain deeper insight into algorithms. For the practical side, demos and examples will usually be done with Matlab, as will some homework assignments. Matlab is easy to learn if you have a programming background.

Grading

The overall course grade is made up of the following components:

Homework ($\sim$ 10 assignments)	42%
Clicker participation	3%
"Team Contest" and Oral Presentation	5% each
Midterm	15%
Final	30%

Exams will be curved before contributions to final grades are calculated.

(The course will make use of iclickers to provide an opportunity to more easily interact and provide feedback in a large class. Answers will not be graded for correctness, though they will be tracked for each student.)

Time Requirements:

The 10 homework assignments each take between 8-15 hours (including recommended readings), with 12 hours per assignment being typical ($\sim$ 120 hours total). In addition, the four weeks of the semester without homework include four additional graded activities, a "team contest", a midterm, an oral presentation, and the final exam; on average, each of these will take 5-10 hours to prepare for ($\sim$ 30 hours total). Class meets for 3 hours per week for 14 weeks (42 hours total).

Learning Goals

The primary goals of this course are to 1) acquire tools to design effective algorithms, and 2) learn to communicate effectively about these sophisticated concepts.

This course surveys the most useful patterns and principles of algorithm design, including: dynamic programming, divide and conquer algorithm design, Fourier transforms, competitive analysis and online algorithms, data structures including techniques for hashing and self-balancing binary search trees, information compression, greedy algorithms, NP hardness, optimization techniques including convex optimization and local search, linear programming and duality, maximum matching and max-flow and related graph algorithms, and a discussion of algorithms in the context of real systems including massively parallel GPU computing.

In the context of the particular material above, students will build a toolset that applies to a broad array of real world problems. In addition, the course develops mathematical sophistication: students will gain the analytical skills to approach new concepts from algorithms and related fields on their own.

Clear thinking and clear communication go hand in hand. This class aims to teach students how to clearly and concisely communicate their thoughts in a collaborative environment, which enables discussions of the sophisticated material. Improved communication includes speaking, writing, and—crucially—listening. Effective communications for the field of algorithms spans from compellingly conveying delicate intuition, to translating this intuition into precise and convincing mathematical arguments.

Goals in Context:

Each homework introduces new algorithmic tools, which students explore and master in the context of a developing body of algorithmic knowledge. Homeworks consist of challenging problems, solved in a collaborative group-work environment guided by a course staff trained to emphasize clear communication. Homework solutions must convey both intuition and formal reasoning, with detailed and separate TA feedback about these two components of the writeup. All homeworks are solved and written with a partner, providing an opportunity to explore and practice new communication styles.

The "team contest" involves solving ~ 20 problems in a group of 4-5 and then participating as each of ~ 20 students from several teams presents the solutions to each other and to course staff. The team contest occurs early in the course to rapidly expose students to many different styles of effective communication. Students practice speaking, listening, and providing constructive feedback in the context of solving a wide range of problems in medium-sized groups. In the "team contest" event itself, students see 20 different effective communication strategies of their peers, along with live interactive feedback and guidance from the course staff. The oral presentation towards the end of the term is a more focused instance of these principles.

Assignments

There will be roughly one assignment due each week in this class. All assignments (unless otherwise stated) will be done in *pairs*. For homeworks 1 through 5 you must use a different partner each time. For the remaining homeworks you may use partners you have used before. Each pair will

turn in a single handin.

Working in pairs will give you an opportunity to improve your thinking, communication, and writing skills. If something you write requires a verbal explanation for your partner to understand it, consider this a valuable sign that this explanation should be included in your writeup. In particular, you are responsible for everything you and your partner submit. It is an academic code violation to sign your name to something that is not yours. Further, however, the material covered on the homework will help prepare you for the exam, so aim for mastery of all of it.

All written assignments must be legible (we prefer they be typed with LaTeX, although we will not penalize hand-written work if it is neat; whether or not you use LaTeX, consider drawing diagrams by hand). Assignments must be handed in to the CS157 handin bin located on the second floor between the fishbowl and the lockers. Each problem should be handed in separately, with your names at the top. Any programs we ask you to write will be handed in electronically using the CS157 handin script, `cs157_handin`.

All homeworks will go out on Friday after class. Most assignments (except homework 0) will have three different due dates: early, on-time, and late. The early deadline will fall on Tuesdays at 6:00 PM. If you turn it in before the early deadline, you will get 5% extra credit. The second deadline is on-time, and it will fall on Fridays at 6:00 PM. The third deadline is late, and it will fall on Sundays at 6:00 PM. A late assignment will result in a 20% deduction. Feel free to turn in different problems at different deadlines, but you must turn in every part of a problem at the same deadline. This holds true even for problems with multiple parts. Sometimes we will ask you to write a program and then, in words, answer some questions about that program. This should be handed in by the same deadline, even though one part is an electronic handin and the other is a paper handin. If you turn in two parts of the same problem at different deadlines, we will grade both parts on the later of the two deadlines.

Because of the fast pace of the class, exceptions/extensions will be granted only in exceptional/extensional circumstances by the professor, and must be requested **at least 24 hours in advance**.

Corrections to the inevitable errors in problem sets will be emailed to the listserv `cs157.2016-17.f@lists.cs.brown.edu`. You are responsible for all information sent to the listserv. If you are registered for the course, you will be added to the listserv at the beginning of the semester. If you are not on the listserv, please email `cs157headtas@cs.brown.edu` and we will add you.

Solutions for homeworks and problem sets will be posted on the course web page, possibly password protected (we will tell you the passwords!).

The midterm will be on **Wednesday, October 19** at 7 – 10PM.

The final (as listed on Banner) will be on **Monday, December 19, 2016** at 9AM – 12PM.

Standards for Written Work

One of the goals of this course, and indeed a computer science education in general, is to train you to write to a professional standard. This means that, unless we explicitly say otherwise, you should justify every answer you turn in, via a proof — of runtime, or correctness, depending on the problem. More generally, you should aim to produce written work which would you could imagine fitting in a computer science journal. Specifically, what you hand in should have a clear order of presentation, with each sentence/algorithm/equation/diagram clearly fitting into a logical

whole. Try reading your homework **aloud** to yourself before you turn it in: if this is impossible, or awkward, then you probably need to organize it differently. As you get better at this, your writeups will become shorter and clearer.

Grading is done by course staff (including undergraduate TAs) under supervision of the professor. We encourage you to discuss homework writeups (past and future) with course staff; bring past homeworks to the TA who graded it, or to more senior course staff—the head TAs and professor.

Oral Presentation and "Team Contest"

Instead of homeworks on two weeks we will have interactive events. Late in the course (the week of November 7), will be an oral presentation, where instead of turning in a written homework, you will solve the problems (individually or in a group) and then you, individually, will sign up for a 10 minute time where the course staff will ask you to explain one or more of your solutions. As a warm up for this, early in the course (week of September 26) we will have a "team contest," a fun event where you will compete on teams to present solutions to a variety of problems under a time limit. These activities are designed to help you build confidence in your ability to clearly present algorithmic concepts, and will complement the written homeworks. The TAs would love to help you improve your presentation skills (written and oral). Come to office collaboration hours.

Contacting the Course Staff

When e-mailing the course staff, please follow these guidelines.

- Do not e-mail individual TAs unless you have a question specific to their grading.
- If you have a quick clarification question or you find that the handin scripts are not working, please e-mail `cs157tas@cs.brown.edu`. To aid us in responding to you quickly,
  - Make sure your subject is a clear, accurate description of the content of your e-mail.
  - Keep your e-mails short. If your e-mail is long (say, more than one screen long), your question is better suited for Collaboration Hours; you should ask your question there.
- Questions or comments about course policies, or more confidential questions should be sent to `cs157headtas@cs.brown.edu`.
- Requests for an extension or questions about final grades should be sent to the professor `pvaliant@cs.brown.edu`.

Your peers or office collaboration hours are usually the best way to get an answer to your question.