Course Description: 6.852, Distributed Algorithms

1 People and places

Instructor: Prof. Nancy Lynch, NE43-365, MIT extension 3-7225, lynch@theory.lcs.mit.edu.
Available after class and by appointment.

Teaching assistant: Rodrigo Rodrigues, NE43-527, MIT extension 3-6015, rodrigo@csail.mit.edu
Office hours: W4-6; NE43-527

Course secretary: Joanne Talbot Hanley, NE43-366, extension 3-6054

Class meetings: Tuesdays and Thursdays, 9:30-11:00AM in room 4-149.

Web site and mailing list: We will set up a course mailing list. Please send email right away to Rodrigo, at rodrigo@csail.mit.edu, to join this list.

The course Web site is at URL http://theory.lcs.mit.edu/classes/6.852/03. This site will contain downloadable copies of course handouts and related research papers, and other information about the course.

2 What this course is about

The field of distributed algorithms has become a well-developed research area over the past 20-25 years. Its results appear in specialized research conferences such as PODC and DISC, and in general conferences involving distributed computing. Distributed algorithms researchers define models for various kinds of distributed computing environments; define problems to be solved in those environments, including relevant cost measures and fault-tolerance requirements; design new distributed algorithms to solve these problems; and analyze the correctness and performance of these algorithms. Distributed algorithms researchers also prove impossibility results and lower bounds, which explain why certain tasks cannot be carried out in distributed settings, or cannot be carried out with a certain cost. Researchers typically study problems that arise in practical distributed systems, such as problems of synchronization, resource allocation, distributed agreement, and management of shared data. Because it is a branch of theoretical computer science, the work is supposed to be mathematically rigorous; however, preliminary papers in this area are often written quite informally.

6.852 is a graduate-level course that is intended to do two things:

1. Provide an introduction to the most important basic results in the area of distributed algorithms.
2. Prepare interested students to begin independent research or take a more advanced course in distributed algorithms.

Usually, the students who take the course are a mixture of PhD students and MEng students. The course is run at a graduate level, which means that most MEng students will find it challenging and time-consuming.

3 Prerequisites

To take 6.852, you should have:

- "Mathematical maturity". In particular, you should be very good at reading and writing mathematical proofs.
- General knowledge about distributed systems. For instance, MIT's undergraduate course 6.033, Computer Systems Engineering, would be good background.
- Experience with sequential algorithms and their analysis. MIT's undergrad course 6.046 is sufficient.
- (Desirable, but not essential) Experience with formal models of computation. MIT's course 6.045 on automata theory would be fine for this.

4 Source material

The main source will be the book *Distributed Algorithms* by Nancy Lynch. See the course Web site for some information on finding and purchasing the book. Quantum Books has ordered copies for us.

The latest printing of the book is the fifth printing, but this is identical to the fourth printing. The fourth printing contains only a few corrections over the third printing, so using a third printing copy would be OK too. Known errata for early printings are collected in errata lists, which are accessible from the course Web page. The book refers to many papers from the research literature on distributed algorithms; you might want to track down and read some of these.

Other books that you might find useful are:

- Hagit Attiya and Jennifer Welch *Distributed Computing: Fundamentals, Simulations, and Advanced Topics*. McGraw-Hill, 1998.
  This is another textbook on distributed algorithms, published a little after the Lynch book. The material covered overlaps quite a lot with the Lynch book, though Attiya and Welch do cover some topics, like clock synchronization, that Lynch doesn't cover. The style of Attiya and Welch's book is less formal. We have put two copies of this book on reserve in the LCS Reading Room in NE43.
- Shlomi Dolev. *Self-Stabilization*, MIT Press, 2000. This gives a good description of self-stabilizing distributed algorithms. Self-stabilization is one kind of fault-tolerance, which we will study near the end of the course.

In addition, some research papers that are not covered in the textbook will be covered in class and on problem sets. They cover a variety of topics, most notably, computability and relative computability results for different kinds of objects and services, in the presence of various numbers of failures. These papers are listed in a separate handout, Handout 3. We will put instructions for obtaining these on the course Web site.

5 Course requirements

5.1 Readings

Readings that cover the material for each class will be announced ahead of time. For most classes, these will be sections from the textbook. For some classes, however, these will be research papers from the reading list, Handout 3. Reading a research paper will generally take more time than reading material from the textbook—be prepared to put in the extra time.

Students are expected to read the assigned material, and to come to class prepared with questions and discussion ideas.

5.2 Problem sets

These are intended to help you to understand the material being covered in class. Most problems will involve thinking about algorithms and problems already covered in class; some will be designed to get you started thinking about ideas to be discussed in later classes.

Specifically, approximately five problems will be given out approximately every Thursday. The problems will be batched and due every two weeks, at the beginning of class on alternate Thursdays. (Note exceptions to these rules on the course schedule, Handout 2.) There will be a total of seven problem set due dates.

No late homeworks will be accepted.
In case of an emergency, please talk to either Rodrigo Rodrigues or Prof. Lynch.

Homework is an important part of your grade. When grading homework problems, we will try to give full credit to solutions that include all the important logical steps and ideas. We consider it a minus for a writeup to be lengthy and overly detailed. An exception is when we specifically ask for details, for instance, in formal proofs of correctness of algorithms.

Solutions to homework problems will be handed out. Whenever possible, the best student solution will be used. Students who would like their writeups to be handed out can help us by writing elegant and concise solutions and formatting them using LaTeX (and the LaTeX style files provided by us). When you submit the homework, keep the .tex file since it may need to be edited if your solution is chosen. Problem sets will be graded by teams of students in the class, led by Rodrigo and/or Prof. Lynch.

Policy on homework collaboration: You are *strongly encouraged* to discuss possible solutions with other class members. Many students in past incarnations of this course have formed homework discussion groups. However, *you must always write up the solutions entirely on your own.*

5.3 Problem set grading

For each problem set, a group of about 5 students will be responsible for working with the course staff to grade the solutions. We would like the grading to be completed by the Monday afternoon after the homeworks are handed in, so we can record the grades and hand them back on Tuesday. The number of times you have to grade over the course of the semester will depend on the size of the class. Part of your grade will depend on the quality and promptness of your work on problem set grading.

5.4 Exams

There will be no exams. No midterm, no final. You can go home after the last class, on December 9.

5.5 Course grade calculation

Your course grade will be based on problem set grades, problem set grading grades, and class participation. Here is how your grade will be calculated:

- Problem sets: 70% (10% for each problem set)
- Class participation (attendance, quality and quantity of participation): 20%
- Grading (quality and promptness): 10%