

COURSE INFORMATION

Professor: Franklyn Turbak (please call me "Lyn")
Office: SCI E106
Phone: x3049
E-mail: fturbak@wellesley.edu
Lectures: SCI E211, Wednesday/Friday, 9:50am -- 11:00am
Office Hours: Tuesday: 4:00pm - 6:00pm
Wednesday: 3:00 pm - 5:00 pm (only 4:15pm on Wednesdays when
there is a computer science colloquium in E101)
Thursday 1:30 pm - 3:30 pm
(Appointments can be made for other times.)

COURSE OVERVIEW

Algorithms are recipes for solving computational problems. In CS231, we will study fundamental algorithms for solving a variety of problems, including sorting, searching, graph algorithms, compression, and cryptography. Even more important, we will focus on general design and analysis techniques that underly these algorithms: e.g., divide-and-conquer, dynamic programming, greediness, probabilistic approaches, and amortized analysis. With an understanding of these techniques, you will be prepared to design some of your own algorithms.

Algorithms are judged not only by how well they solve a problem, but also by how effectively they use resources like time and space. We will learn techniques for analyzing the time and space complexity of algorithms and will use these to evaluate tradeoffs between different algorithms. We will also see that problems can be organized into a hierarchy that measures their inherent difficulty by the efficiency of the best possible algorithm for solving them.

PREREQUISITES

One prerequisite for CS231 is CS230, *Data Structures*. You should feel comfortable with simple data structures (arrays, lists, trees, stacks, queues) as well as recursion.

The other prerequisite is a willingness to learn some math. In order to analyze algorithms, we'll need to use some mathematical methods that may be new to you. We will learn these methods in class, but they are also explained well in the textbook.

The notation we will use for algorithms is the pseudocode used in the textbook. We will not do any programming in this class.

TEXTBOOK

Required: *Introduction to Algorithms*, by Cormen, Leiserson, and Rivest (1990). (Henceforth, I will refer to this book as CLR.) This excellent and comprehensive textbook is a “must” for any computer scientist. It’s a thick book, but don’t worry --- we’re not going to cover the whole thing this semester! Most chapters are short and have good explanations, examples, and exercises. However, a few topics are somewhat “dense”; we’ll spend some time unravelling these.

The book is expensive (\$67.00), but is one of the few textbooks that may actually be worth what it costs. You may be able to find a cheaper used copies at the Wellesley bookstore or at the MIT Coop. Another place to look is the “hurt books” section of the MIT Press Bookstore (which is right across the street from the MIT Coop). You may be able to borrow or buy a copy of the book from someone who took CS231 last fall. Finally, for those who may not wish to purchase the textbook, I will put a copy of it on reserve at the Science Center Library.

Other Books: I have put several other algorithm books on reserve at the Science Center Library. I encourage you to consult these for alternative explanations of the material in the textbook:

- Alfred v. Aho, John E. Hopcroft, Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley (1974).
- Sara Baase. *Computer algorithms : introduction to design and analysis* Addison-Wesley (1988).
- Ellis Horowitz and Sartaj Sahni. *Fundamentals of Computer Algorithms*. Computer Science Press, 1984.
- Donald E. Knuth, *The Art of Computer Programming, Volume 3: Sorting and Searching*, Addison-Wesley (1968). This is the *classic* reference on sorting and searching.
- Robert L. Kruse, *Data Structures and Program Design* (3rd edition, 1994). This CS230 textbook has a number of good sections on advanced algorithms not discussed in CS230. (Note: This is a CS230 reserve, not a CS231 reserve.)
- Robert Sedgewick, *Algorithms*, Addison-Wesley (1983). A popular textbook that was used for many years in CS231. Uses Pascal as the language for expressing algorithms.

ANIMATIONS

Thanks to an algorithm animation project at Dartmouth, there are animation tools for many of the algorithms in CLR. You can access these from the `Animation` subfolder of the CS231 folder on Kimba (see below for more info on the course folder).

HOMework

There will be weekly problem sets that involve modifying, analyzing, and designing algorithms. All of these problem sets will be of the paper and pencil variety; no programming will be involved.

All problem sets are due in class on the advertised due date, which will typically be a Friday. Problem sets will be graded on a 100 point scale. I will strive to have problem sets graded by the next class meeting after the due date. At this time, solutions will be distributed with the graded homework.

There will be **ten** problem sets during the term, but I will only count your **eight** best assignments when calculating your final grade. This gives you the option to “punt” two problem sets during the term; choose these wisely! However, you are responsible for knowing the material on all of the problem sets.

Problem sets will include three kinds of problems:

- **Suggested problems** are problems that you should think about, but you should *not* write them up and turn them in. They will typically be exercises from the book.
- **Required problems** are problems whose solutions you must write up and turn in.
- **Extra credit problems** are optional problems whose solutions you may write up for extra credit.

Your problem set grade will depend only on the required problems. However, doing the suggested problems will help you understand the material better, and you will not have to spend any time writing them up. As motivation for doing the suggested problems, I offer the following pledge: **at least one problem on each exam will be a suggested problem from one of the problem sets.**

To make up for points lost on problem sets and exams, students often request extra credit problems. In order to give everyone the same opportunity, I will often include extra credit problems on the problem sets. The extra credit problems are often more difficult than the other problems, but they provide the opportunity to earn extra points toward your course grade. You should only attempt extra credit problems after completing the assigned problems.

Extra credit problems are entirely optional. Extra credit points count as problem set points, but they are only factored into course grades *after* I have partitioned the grade scale into letter grades. This means that doing the extra credit problems may raise your course grade, but not doing extra credit problems will not lower your course grade.

For maximum flexibility, **you may turn in extra credit problems at any time during the term** (through the end of finals week). However, experience has shown that students who leave extra credit problems until the end of the term rarely turn them in. It is in your best interest to complete extra credit problems in a timely fashion. I will not hand out solutions to extra credit problems, but you are encouraged to discuss them with me in person.

PROBLEM SET HEADER SHEETS

I would like to get a sense for how much time it takes you to do your CS231 problem sets. Please keep track of the time you spend on each problem of your problem sets, and include this information on the problem set header sheets that I will provide at the end of each problem set. Turn in this header sheet as the first page of your hardcopy solutions.

LATE HOMEWORK POLICY

I realize that it is not always possible to turn in problem sets on time. On the other hand, turning in one problem set late can make it more difficult to turn in the next problem set on time. I have decided on the following policy for this course this term:

Problem sets due on Friday will be accepted until 6pm on Saturday. After this time, they will *not* be accepted, so turn in whatever you can by then.

The fact that two problem sets are not counted (see previous section) softens this policy somewhat. You may turn in late problem sets by slipping them under my office door. Please write the time of submission at the top of a late problem set. I rely on the honor code to guarantee that no submissions are made after 6pm on Saturday.

In extenuating circumstances (e.g., sickness, personal crisis, family problems), you may request an extension without penalty. Such extensions are more likely to be granted if there are made *before* the due date.

COLLABORATION POLICY

Since I believe that collaboration fosters a healthy and enjoyable educational environment, I encourage you to talk with other students about the course and to form study groups.

Unless otherwise instructed, feel free to discuss problem sets with other students and exchange ideas about how to solve them. However, there is a thin line between collaboration and plagiarizing the work of others. Therefore, I require that **you must compose your own solution to each assignment**. In particular, while you may discuss problems with your classmates, you must always write up your own solutions from scratch. It is **unacceptable** for two students to turn in copies (or near copies) of each other's solutions.

In keeping with the standards of the scientific community, **you must give credit where credit is due**. If you make use of an idea that was developed by (or jointly with) others, please reference them appropriately in your work. E.g., if person *X* gets a key idea for solving a problem from person *Y*, person *X*'s solution should begin with a note that says "I worked with *Y* on this problem" and should say "The main idea (due to *Y*) is ..." in the appropriate places. It is **unacceptable** for students to work together but not to acknowledge each other in their write-ups. The problem set header sheet includes a spot to list your collaborators.

When working on homework problems, it is perfectly reasonable to consult public literature (books, articles, etc.) for hints, techniques, and even solutions. However, you must reference any sources that contribute to your solution. Assignments and solutions from previous terms of CS231 are **not** considered to be part of the "public" literature. You must refrain from looking at any solutions from previous terms of CS231 (unless, of

course, I explicitly tell you it's OK to do so). It is my policy that consulting previous assignment solutions constitutes a violation of the honor code.

EXAMS

There will be three exams, all open book and open notes:

1. A take-home exam that will be handed out on Friday, October 4 and will be **due at the beginning of class on Friday, October 11.**
2. A take-home exam that will be handed out on Friday, November 8 and will be **due at the beginning of class on Friday, November 15.**
3. A final exam during the regular exam period.

Please mark these dates in your calendars. **You are not allowed to collaborate with anyone else on the take-home exams.**

GRADES

The course grade will be computed as shown below:

Problem sets (total)	40%
Take-home exam 1	20%
Take-home exam 2	20%
Final exam	20%

The default ranges for grades are expressed as a percentage of total points (excluding extra credit points):

A	93.33 -- 100
A-	90 -- 93.32
B+	86.66 -- 89.99
B	83.33 -- 86.65
B-	80 -- 83.32
C+	76.66-79.99
C	73.33-76.65
C-	70 -- 73.32
D	60 -- 69.99
F	below 60

I reserve the right to lower boundaries between grades, but I will not raise them. This means that I can grade on a curve, but only in your favor.

The above information is intended to tell you how I grade. It is not intended to encourage a preoccupation with point accumulation. You should focus on learning the material; the grade will take care of itself. If you are dissatisfied with the grade you will receive based on the above scale, I encourage you to turn in extra credit problems to raise your grade.

COURSE FOLDER

The CS231 course folder is located on Kimba, a file server accessible on the Lucy AppleTalk Zone. This folder contains material relevant to the class, including algorithm animations and on-line copies of the handouts.

COURSE BULLETIN

There is a CS231 folder in the Bulletin system on Lucy. This folder has several purposes. I will use it to make class announcements, such as corrections to assignments and clarifications of material discussed in class. I encourage you to post questions or comments that are of general interest to the course. The course bulletin is also a good place to find people to join a study group. Plan to read the CS231 bulletin on a regular basis. Use the `set brief CS231` command to indicate that you want to be notified when there are new CS231 messages.

FINDING HELP

If you have any questions at all about the class (whether big or small, whether on problem sets lectures, reading, or whatever) *please* contact me. **That's what I'm here for!**

The best time to see me is during my scheduled office hours (which are listed at the top of this handout). If these times are not convenient, we can set up an appointment at some other time. You can set up an appointment by talking with me in person, calling me on the phone, or sending me email. You can also ask questions by sending me email. I read my email on a regular basis, and will check it even more frequently in the few days before an assignment is due.

Drop-in tutors are available to answer your questions during certain hours. The names and schedules of the drop-in tutors will be made available early in the term. If you are having trouble with the course, you can request a one-on-one tutor from the Learning and Teaching Center (LTC) . This service is confidential and free of charge; please take advantage of it if you would like some extra help! Contact me or LTC for more information about this service.

Finally, when looking for help, don't overlook your fellow students --- not only those who have taken the course in the past, but your classmates as well. Your classmates are a valuable resource; make good use of them!

FEEDBACK

I am eager to hear your feedback on the course! You can talk to me in person, send email, or drop an anonymous note into the Suggestion Box in the CS231 folder on Kimba.

STUDENTS WITH SPECIAL NEEDS

If you have any disabilities (including "hidden" ones, like learning disabilities), I encourage you to meet with me so that we can discuss accommodations that may be helpful to you.