CECS 428 Syllabus

Darin Goldstein*

1 Contact info

- Office: ECS 546
- Phone: (562)985-1507
- Email: daring90808@gmail.com
- Office hours: 12:30-2:00 PM TTh
- Web page: http://www.csulb.edu/∼dgoldst2/
- Term: Fall 2016
- Class Meeting Times: Lecture 10:00-10:50 MW, Lab 11:00-12:15 MW (though lecture will be given during the lab time)
- Class Location: VEC 518/ ECS 403
- Textbook: There are at least two books for this course:

  Approximation Algorithms for NP-Hard Problems by Dorit Hochbaum and Online Computation and Competitive Analysis by Allan Borodin and Ran El-Yaniv. Lecture notes will be made available online.

2 Objectives

This course is meant to follow CECS 328. In CECS 328, the goal was to identify certain algorithmic tricks which, when applied cleverly to tractable problems, yield "good" solutions. We also learned certain analysis methods which helped us to quantify exactly how good the algorithm was. At the end of 328, we identified a set of problems which, though easy to state, seemed very hard to find good algorithms for.

In this course, we will continue where we left off in 328. We will start by learning one final analysis method and then move on to find out that certain problem do indeed look hard (though how hard, we are embarrassed to say we do not know). We will then learn multiple ways of dealing with these problems. Just because a problem is hard doesn't mean we don't need a solution to it...

* *daring90808@gmail.com*

2.1 Stuff you should already know

- Mathematical maturity: how to evaluate arithmetic and geometric sums, log rules, limits, derivatives, integrals, Lagrangians, elementary probability, etc.
- Computer science maturity: Absolutely everything from Honors CECS 228/328 including modular arithmetic, MSTs, network flows, NP-completeness, etc.

3 How this course will be graded

There will be a total of 8 programming assignments in this course, each worth an equal portion of the final grade.

4 Lectures

Some good news for this course is that you will never be explicitly penalized for missing a lecture. I will never give any pop-quizzes. All graded material is mentioned explicitly somewhere in this syllabus. The final exam date for this course is set by the University (totally independently of me and over which I have absolutely no control) and should be available via the University website.

On the other hand, this is definitely a lecture-based course. If you choose to miss a lecture, I will not penalize you for it or hold it against you in any way, but you are fully responsible for any material that I go over. If I mention something in lecture that is not in the book, YOU ARE STILL RESPONSIBLE FOR KNOWING IT. I will not redo a lecture for people that missed it the first time. It is your responsibility to get the notes/information. If you miss any kind of instructions about assignments that are given during lecture (including but not limited to due dates, methods for submission for assignments, etc.), it is STILL your responsibility to be aware of what occurred in lecture.

5 Withdrawal policy

The University allows a student to withdraw from a class up until a certain date for a "serious and compelling reason" with a professor's signature. I will sign drop forms that list the following reason: "Inability to complete the coursework." For this course, that will be the instructor's interpretation of a serious and compelling reason.

6 Programming Assignments

All programming assignments are to be written in Java.

When it comes to questions about the assignment: I will answer any question pertaining to what it is you have to do and no questions about how to do it.

You may not consult with anyone other than me and your textbook about the programming assignments unless otherwise explicitly allowed by me via email. I will never write code for anyone in the class. All coding will be completely left up to you, and you may not use code written by anyone other than you unless authorized by me via email. IF I FIND OUT THAT COLLABORATION HAS OCCURRED ON ANY PROGRAMMING ASSIGNMENT, THE PENALTY IS AN AUTOMATIC F IN THE COURSE FOR ALL PARTIES INVOLVED REGARDLESS OF WHO DID THE ACTUAL WORK. This policy will be rigidly enforced. It is your responsibility to make sure that nobody is able to copy your work. I recommend never leaving a lab computer unattended, never letting anyone see your code, and always deleting your work from any public computer. (If you need a place to save your work, e-mail it to yourself.) If you suspect that your work is being copied (or cheated off of in any way) by someone else, you may let me know by e-mail BEFORE you turn in the assignment and I will take it into account. However, I will not take into account statements about cheating that do not explicitly identify the cheating party.

I reserve the right to call anyone into my office hours or lab time for any reason to explain any issues with your source code (e.g. if your code does not produce an answer that you submitted, if your code too closely matches that of someone else, etc.). Failure to show up for such a meeting will certainly earn a 0 on the assignment and most likely an F in the course depending on the circumstances (e.g. if I suspect cheating is involved). If a student is unable to adequately explain his/her work on a programming assignment, the grade on the assignment may be reduced appropriately.

Note that there are two types of assignments. Either you will have to (a) have your code solve increasing difficult problem instances or (b) resubmit answers until you reach the grade that you are satisfied with. Almost always, you will need to resubmit multiple times for any given assignment.

The programming assignments will be graded as follows: Each assignment is posted online and you may demonstrate the assignment at any time *before the deadline*. Once the deadline arrives and you have not demonstrated, your project will receive no credit. You will generate and submit your assignments via the server as follows.

1. Log into the server and enter your section and the assignment you're interested in.

2. Generate an instance of the problem. Your individual problem will arrive to you via email in an attachment named input.zip. (Make sure that your Beachboard emails are up-to-date. Note that if you add the course after the semester has officially begun, you will need to explicitly inform me via email that you are unable to log into the server so that I can add an account for you.)

3. Solve the problem and generate an answer. Zip up your answer and rename

it to output.zip.

4. Log back into the server and submit your answer. Once your answer is checked and verified, you will be notified by email. (Note that you may have to do this step several times.)

5. Once you are satisfied with your grade and have generated and solved as many instances as necessary, you will zip up all of your source files into a single zip file named src.zip. (All that is required is to go into your Java project and literally zip up the src folder.) You will then submit it via the server page. *Do not bother emailing code to me!* Only .java files will be accepted. If your scr folder contains anything but .java files, your submission will be rejected. Any assignment that doesn't have a valid code submission by the deadline will be counted as a 0.

The programming assignments will be graded on a scale of 0 to 4. (You can think of a 4 as an A, 3 as a B, etc.)

IF YOU LEAVE YOUR DEMONSTRATION FOR THE LAST MINUTE, IT IS VIRTUALLY CERTAIN THAT YOU WILL NOT BE ABLE TO GET ANY CREDIT FOR THE ASSIGNMENT!!! The server can only handle a certain load at any given time, and if you leave your demonstrations for the last minute, the server may be unable to check your submission. This is *not* the fault of the server, and you will not receive any credit nor will any extensions be granted based on the inability of the server to check your assignment. It is *your* responsibility to budget the time necessary to check your solutions! Plan ahead or plan to not get credit.

If you have any questions or comments about the rules, it is your responsibility to let me know about them IMMEDIATELY so that they can be straightened out.

Requests for late submissions will not be entertained.

Your grades will be available on Beachboard as soon as possible after grading has been completed.

7 Cheating

Cheating on any graded material in the course will lead to an automatic grade of F in the course. The University Honors committee will also be notified and ejection from the Honors program is possible/probable. I don't give warnings.

8 The final word

If there is anything on this page that you have a question or comment about, it is very important to let me know about it on the FIRST DAY OF CLASSES. After the first day of classes, I will assume that you are aware of the grading policies. Any grading misunderstandings you have after the first day of classes are your responsibility. Good luck.

9 Course topics

The programming assignment is due the day of the lecture above.

1. Amortized analysis: The pancake stack
2. Amortized analysis: Dynamic tables, Binary counters
3. Binomial Heaps
4. Fibonacci Heaps I
5. Fibonacci Heaps II
6. Disjoint Sets I
7. Disjoint Sets II
8. Disjoint Sets III
9. The good subspace: Simple problems
10. The good subspace: Satisfiability I
11. The good subspace: Satisfiability II
12. The good subsapce: Vertex cover in bipartite graphs
13. Fixed parameter tractability: 0-1 knapsack problem, Vertex cover problem
14. Fixed parameter tractability: Planar independent set
15. Simple approximations
16. Approximations: Metric travelling salesman problem
17. Approximations: Bin-packing
18. Approximations: Metric k-center problem I
19. Approximations: Metric k-center problem II
20. Randomized approximations: Maximum satisfiability
21. Simple Las Vegas Algorithms: Randomized quicksort
22. Simple Monte Carlo Algorithm: Karger's
23. Online problems: The file cabinet I
24. Online problems: The file cabinet II
25. Online problems: The file cabinet III
26. Splay Trees I: Analysis

27. Splay Trees II: Information
28. Splay Trees III: Kraft Inequality
29. Splay Trees IV: Asymptotic static optimality

10 Assignment due dates

Because nothing needs to be turned in by hand, all assignments are due on a Friday.

1. 09.2.2016
2. 09.16.2016
3. 09.30.2016
4. 10.14.2016
5. 10.28.2016
6. 11.11.2016
7. 11.25.2016
8. 12.09.2016