CSE 541T Advanced Algorithms January 11, 2019

Course Overview and General Information

Your primary source for class information, homeworks, and handouts is the class web site, `https://classes.engineering.wustl.edu/cse541t`. Please check this site regularly for course announcements. You can send me email at **jbuhler@wustl.edu**. Ongoing discussions and announcements may be found in the class Piazza board at `https://piazza.com/wustl/spring2019/cse541/home`. Homeworks will be turned in and Grades will be posted via Canvas at `https://mycanvas.wustl.edu`.

- **Where and When**: Mondays and Wednesdays 2:30-4:00 PM in Brown Rm 118.
- **Prerequisites**: CSE 241 or 247+347, or equivalent undergraduate training in algorithms and data structures, complexity analysis, and basic proof techniques. Some prior experience is assumed with greedy algorithms and dynamic programming.
- **Your Instructor**: Dr. Jeremy Buhler, *jbuhler@wustl.edu.*
- **Your Teaching Assistants**:
  - Zihao Deng
  - Alex Durgin
  - Aaron Kirtland
  - Adrien Xie
- **Office Hours**: My hours and the TAs' hours are posted to the course Piazza board. If you want to meet me outside my hours, please drop me an email.
- **Textbook**: Cormen, Leiserson, Rivest, and Stein's *Introduction to Algorithms*, 3rd Edition. Earlier editions are not sufficient because I may assign problems out of the book that only appear in the latest edition.

1 Course Philosophy and Structure

This course is about the design and analysis of algorithms, with a strong emphasis on *optimization.* That is, given a problem with (possibly) a multitude of feasible solutions and a measure of how good each feasible solution is, we want to find a feasible solution that is the *best* possible by this measure. The problems will be *combinatorial* in nature; that is, they will have discrete (as well as possibly continuous) components.

The only way I know to achieve proficiency in solving combinatorial optimization problems is through experience, i.e. by thinking your own way to a solution. My goals are to point out some basic structures that are common to large classes of optimization problems and, with your help, to work through examples that show how these structures can be exploited to produce efficient algorithms (or to recognize that no such algorithms are likely to exist).

Seeing a solution is only half the battle in algorithm design; you must also express your solution in a form that clearly communicates it and *convinces others (and yourself!) of its correctness.* The standard of evidence for correctness in this course is a *formal proof*, just as you would encounter in any upper-level mathematics course. In this course, I want everyone to improve their competence at explaining and justifying algorithms in the language of formal proof.

Learning to write a good proof requires guidance and practice. You will get some of both from the class meetings and the homeworks, respectively, but you will also have the opportunity to work interactively on proof writing through a weekly *recitation section*, led by Alex and Zihao. Attendance at recitation section, which will run 1-1.5 hours, is not mandatory but *is* highly recommended, especially if you are looking to build your proof skills. These sessions will be run a bit like a "studio" for proof-writing. Participants will first divide into small groups to work on combinatorial optimization problems. Following this small-group work, there will be an opportunity for a subset of groups to present their correctness proofs and for others to critique them for correctness, clarity, and structure.

I will supply you with some worked examples for each major chunk of the course in the form of "practice problems." The problems and their solutions may be downloaded separately from the course web site. *Please try to work these problems yourself* before looking at the solutions – they are the best way to build and check your understanding. If you come seeking help with the homework, the first thing I'll probably ask is how you've fared with the practice problems.

You should expect to spend at least 10-14 hours on each homework, including time to work the practice problems. For each homework problem, you will need to understand what is being asked, see *how* to apply basic structures to it (e.g. greedy choice, dynamic programming, reduction arguments, approximation), and write both a clear, concise description of your solution *and* a formal proof justifying both its correctness and its time complexity. *Please start early* on the homeworks. Be prepared to put aside some of the problems and come back to them. Steady mental effort, perhaps spaced over a period of hours or days, is usually rewarded. Electronic composition of homeworks is also helpful, since it lets you revise and improve your arguments without spending time rewriting the parts you are satisfied with.

2 Homeworks

There will be four to five homework assignments, which will be distributed in PDF form from the course web page.

2.1 Format and Mechanism

Assignments **must** be turned in electronically using Canvas and **must** be typeset (though you are welcome to hand-draw your figures). We cannot accept assignments submitted by any other mechanism. *Handwritten assignments will be penalized at least 20%*; we may, with suitable notice, change this policy later in the semester so as not to accept such assignments at all. Please see the course website for detailed turn-in procedures and advice on composing your homework electronically.

2.2 Late Policy

Assignments must be turned in by class time on the due date. Late assignments will be not be accepted except by prior arrangement.

I expect your homework solutions to be clear, concise, and easy to read – if the graders and I cannot understand your argument, we will mark it wrong.

2.3 Student Graders

I will be asking for student volunteers to help grade the homeworks. If you have solved a homework problem and want to grade it, you must write up your solution and submit it for review by a private post to the course's Piazza board *before* its due date. Graders will be chosen on a first-come, first-served basis. Due to the difficulty of grading 80+ assignments in a timely fashion, I usually seek 2-3 student graders per problem.

Depending on how well I feel you did at grading a problem, you may receive extra credit up to one-half the value of the problem you grade. This credit is added to your score on the relevant homework. For example, if you grade a problem worth 20% of credit on Homework 3, and your grade on this homework is 92, then your grade with the additional credit would be $92 + 10 = 102$.

I will try to avoid having the same student grade several homework problems during the semester; however, I can only spread the grading credit fairly if everyone takes the time to apply for grading and to provide me with good solutions in advance!

3 Exams

There will be two 80-minute exams held in class during the semester, plus one take-home final. In-class exams are always closed-book and closed-notes. However, you may use one 8.5×11 crib sheet (both sides). Ordinary calculators are permitted, but not graphing or programmable ones.

If you have a documented accommodation for in-class exams, such as extra time, I will ask you to arrange to take the exam in Cornerstone at the same time as the rest of the class.

Exams may be either graded in hard copy or scanned and returned via Gradescope.

4 Grades

Your final grade in the course will be weighted roughly as follows:

1. each homework: 8% (assuming five)

2. each exam: 20%

I reserve the right to curve grades for individual assignments and exams, as well as the final course grade, to adjust for assignments that prove unusually challenging. But I will not curve any grade *down* – your curved grade will always be at least as high as the uncurved grade.

After any curving, the letter-grade thresholds for your final numerical grade will be the usual 90, 80, 70, 60 for the A, B, C, and D ranges.

5 Policy on Collaborations and Academic Integrity

Please see the separate collaboration policy document on the course web site. You are expected to be familiar with this policy and to abide by it at all times. *By turning in an assignment, you certify that you have followed the course collaboration policy for that assignment.*