15-351/15-650/02-613 Algorithms & Advanced Data Structures
Spring 2019

Instructor: Hosein Mohimani, *Office:* GHC 7717, *Email:* `hoseinm@cs.cmu.edu`

Textbook: *Algorithm Design* by Jon Kleinberg and Éva Tardos. Addison Wesley, 2005. ISBN: 0-321-29535-8.

Office hours: Will be posted on the class webpage. You are encouraged to make use of office hours, and the TAs will be as helpful as possible to explain material and homework questions and solutions.

Final: According to the university's schedule

Course objectives: The objective of this course is to study general computational problems and their algorithms, with a focus on the principles used to design those algorithms. After passing this class, you should be able to:

1. analyze running time for many kinds of algorithms
2. design divide-and-conquer algorithms
3. design dynamic programming algorithms
4. design network flow-based algorithms
5. write linear / integer programs
6. apply large-scale search / heuristic algorithms
7. efficiently store and answer queries about data
8. prove a problem is NP-complete

Coursework: Coursework will consist of near weekly homeworks that will include algorithm design and analysis problems and some programming assignments, 2 midterms, and a final. The midterms will be non-cumulative, while the final will cover everything from the class.

Grading: The class will be graded on a curve. Your grade will be computed in two ways, and we will use whichever approach gives you the higher grade:

- Standard grading: 20% for homeworks, 25% for each of the midterms, and 30% for the final.

- Alternative minimum grade: 20% for homeworks, 15% for the lower midterm, 25% for the higher midterm, 40% for the final.

The alternative minimum grading approach attempts to help students who struggle early on, but make it up later or who have a bad day on one of the midterms. However, to be eligible for the alternative minimum grading approach, you must have turned in 90% of the homeworks.

What about programming? This course is **not** a programming course. It is an algorithm design and analysis course. Part of the point of the course is to learn to be able to think about algorithms separately from programming. The course's focus is on being able to understand algorithms theoretically not empirically. You are of course welcome (and encouraged!) to implement any of the discussed algorithms as a way to help yourself understand them, and we can provide starting source code for some of them.

What's the difference between 02-613, 15-650 and 15-351? All the courses have the same structure, same lectures, same TAs, etc. Those who are signed up for one of the graduate numbers (02613 or 15650) will have a few additional assignments. The grading curve will also be computed separately in each of the three courses.

Tentative schedule: The class has 4 major subunits:

1. Introduction, Minimum Spanning Tree case study (with Heaps, Union Find, Graph data structures), Asymptotic analysis [≈ 1 week]
2. Divide & conquer and graph algorithms [3 weeks]
    - Graph search: Breadth first, depth first, topological sorting
    - Finding closest pair of points
    - Fast Fourier Transform
    - Matrix Multiplication
    - Shortest path algorithms
    - A* heuristic search
3. Additional Data Structures [≈ 1.5 weeks]
    - Suffix trees & string matching
    - Splay trees & amortized analysis
4. Advanced algorithmic design techniques [2nd half of semester]
    - Dynamic programming (edit distance, RNA folding, chains of matrix multiplication, etc.)
    - Network flow and its use for solving problems (like matching, survey design)
    - Linear and integer programming
    - NP-completeness
    - Randomized algorithms (hashing & global minimum cut)

Homework policies:

- Homeworks are due at the start of class. They will be submitted through Blackboard. **No late homework will be accepted** — turn in what you have completed. If you will miss class, turn in the homework early.

- Answers to homework problems should be written concisely and clearly. Homeworks must be typeset and submitted as PDFs. Instructions for submission will be posted on the course webpage.

- Homework problems that ask for an algorithm should present: a clear English description or pseudocode, an argument that the algorithm is correct, and an analysis of the running time. Note: your goal is to explain the algorithm to a human, not a computer — as such detailed pseudocode or source code is usually *not* the best way to explain an algorithm.

- Graded exams and other material should be picked up in class; if you miss the class when the homework is returned, please pick it up during office hours.

- Regrade requests should be made **in writing** within 1 week of the homework being returned. The entire homework or exam in question will be regraded, which may result in a higher or lower grade than originally returned.

- You may discuss homework problems with classmates. You must list the names of the class members with whom you worked at the top of your homework. **You must write up your own solution independently!** "Independently" means — at least — that you cannot look at another person's homework, you cannot have them look at yours to see if it is correct, you cannot take detailed notes from a discussion and edit them into your homework, and you cannot sit in a group and continue discussing the homework while writing it up. The intent of this rule is: you can gather around a whiteboard with your fellow students and discuss how to solve the problems. Then you must all walk away and write the answers up separately. Note: it's really the exams that count for most of your grade, so there's little benefit in writing down a homework answer that you don't understand.

  Unfortunately, each semester, we find some people who have copied each other's homeworks. Such instances are referred to the University according to the academic integrity violation policy.

- You may *never* use, look at, study, or copy any answers from previous semesters of this course.

- You must write all programming assignments on your own and cannot share code with other students or use code obtained from other students. In addition to manual inspection, we use an automatic system for detecting programming assignments that are significantly similar.

Classroom etiquette: To minimize disruptions and in consideration of your classmates, I ask that you please arrive on time and do not leave early. If you must do either, please do so quietly. Laptop use is discouraged — their use detracts significantly from the benefit of coming to class (wouldn't it have been more fun to spend an hour surfing Facebook at home?) and also provides a distraction for other students. If you must use your laptop, please turn the sound off, type quietly, and sit as far towards the back of the room as possible.

Excused absences: Students claiming an excused absence for an in-class exam or midterm must supply documentation (such as a doctor's note) justifying the absence. Absences for religious observances must be submitted by email to the instructor during the first two weeks of the semester.

Academic honesty: All class work should be done independently unless explicitly indicated on the assignment handout. You may *discuss* homework problems with classmates, but must write your solution by yourself. If you do discuss assignments with other classmates, you must supply their names at the top of your homework / source code. No excuses will be accepted for copying others' work (from the current or past semesters), and violations will be dealt with harshly. (Getting a bad grade is much preferable to cheating.)

The university's policy on academic integrity can be found here: `http://www.cmu.edu/policies/documents/AcademicIntegrity.htm`. In part it reads "Unauthorized assistance refers to the use of sources of support that have not been specifically authorized in this policy statement or by the course instructor(s) in the completion of academic work to be graded. Such sources of support may include but are not limited to advice or help provided by another individual, published or unpublished written sources, and electronic sources." You should be familiar with the policy in its entirety.

In particular: use of a previous semester's answer keys or online solution manuals for graded work is absolutely forbidden. Any use of such material will be dealt with as

an academic integrity violation.

Frequently Asked Questions

Is there some extra work I can do to improve my grade?

No, we cannot make exceptions to the course work and grading policy. If you are concerned about your grade, please seem me or one of the TAs ASAP. There will be no exceptions to this policy during or after the class has completed.

I have to be out of town, and I would like an extension on my homework. Can I have one?

No. This is a very large class, and it is not possible to accommodate individualized deadlines for everyone. You can always turn your homework in early (if the link is not available, please ask a TA). You can also always turn your homework in remotely since we will use Blackboard to submit homeworks.