

02-713 Algorithms & Data Structures for Scientists Spring 2013

- Instructor:** Carl Kingsford, *Office:* GHC 7705, *Phone:* 412-268-1769, *Email:* carlk@cs.cmu.edu
- TAs:** David Farrow, *Office:* GHC 7603 *Email:* dfarrow@andrew.cmu.edu
Geet Duggal, *Office:* GHC 7413 *Email:* geet@cs.cmu.edu
- Web page:** <http://www.cs.cmu.edu/~ckingsf/class/02713-s13/>
- Class time:** MWF 9:30-10:20am in HH B103
- 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
- Midterms:** In class on Friday, March 1st, 2013 and Friday, April 26, 2013
- 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 some kinds of queries about data
8. prove a problem is NP-complete

Course work: 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. Approximate grading weight: 10% for homeworks, 25% for each exam, and 40% for the final. The class will be graded on a curve.

What do you mean “for scientists?” The goal of this course is to cover many algorithms and algorithm design techniques. The topics we cover are not substantially different than you would see in any other algorithm class. We use a standard algorithms textbook. However, the course is tailored “for scientists” in two ways: (1) we don't assume a large CS background, but rather only the mathematical maturity & logical thinking that any scientist would have, and (2) when there's a choice, we choose example applications and problems that are interesting to some scientific field.

Tentative schedule: The class has 3 major subunits:

1. Introduction, Minimum Spanning Tree case study, and Python [2 weeks]
2. Elementary algorithms: divide & conquer and graph algorithms [4 weeks]
 - Asymptotic analysis
 - Closest pair of points
 - Fast Fourier Transform
 - Graph search: Breadth first, depth first, topological sorting
 - Shortest path algorithms
 - A* heuristic search
3. Advanced algorithmic design techniques [2nd half of semester]
 - Dynamic programming
 - Network flow
 - Linear and integer programming
 - NP-completeness
 - Randomized algorithms

Homework policies:

- Homeworks are due at the start of class. **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.
- Graded homeworks 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.
- 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: This is a big class. 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 cheating and plagiarism can be found here:

<http://www.cmu.edu/policies/documents/Cheating.html>. In part it reads "In any presentation, creative, artistic, or research, it is the ethical responsibility of each student to identify the conceptual sources of the work submitted. Failure to do so is dishonest and is the basis for a charge of cheating or plagiarism, which is subject to disciplinary action." You should be familiar with the policy in its entirety.