Algorithms (CS 300)

Lecturer:
Otfried Cheong

Lecture time: Mon, Wed 13:00–14:15

Course webpage: http://otfried.org/courses/cs300

250 Students tried to register for CS300A.

After the lottery, more than 100 students sent me email asking to be added to the course.

I cannot work miracles. I would have offered a second section of CS300, but I'm already teaching two courses this semester.

If you are not registered for the course, and if you did not receive an email from me offering to add you to the course, then you cannot take CS300A this semester. Sorry.

CS300 is offered every semester, and we will provide sufficient capacity in the fall semester.

Again: If you are not registered for CS300A and did not get email from me promising to add you, then you can go now.

Textbook

Algorithms by Dasgupta, Papadimitriou, Vazirani. A good draft used to be available on-line and there is a Korean translation.

Also lecture notes by Jeff Erickson and Avrim Blum.

Homework

Many small theoretical homeworks on paper. You will have about one week for each homework.

Programming homeworks

There will be one or two graded programming homeworks. You must submit all programming homeworks.

Acceptable programming languages are Python, Kotlin, Scala, Java, C, C++.

Grading Policy

Programming Homework (10%), Paper Homework (10%), Midterm (30%), Final (40%), Participation (10%);

Attendance

We will take attendance in every lecture. You have three missed lectures free—use this for doctor appointments, interviews, etc. You do not need to send me email about missing a lecture.

Exams

The exams are on April 16 and June 11, during the exam weeks, from 13:00 to 15:45.

Classum

We will try out Classum, a new platform for course announcements and Q&A.

Since important announcements will only be made on Classum, every student must make an account on www.classum.org and register for CS300A.
You must then regularly check the announcements on Classum.

We will also use Classum for answering all your questions about the course contents. You can ask questions anonymously. You can ask questions in English or Korean.

What is this class about?

This course introduces basic concepts of design and analysis of computer algorithms: the basic principles and techniques of computational complexity (worst-case and average behavior, space usage, and lower bounds on the complexity of a problem), and algorithms for fundamental problems. It also introduces the area of NP-completeness.

Etymology of "Algorithm"

Algorism = process of doing arithmetic using Arabic numerals.

A misperception: algiros [painful] + arithmos [number].

True origin: Abū 'Abdallāh Muhammad ibn Mūsā al-Khwārizmī was a 9th-century Persian mathematician, astronomer, and geographer, who wrote Kitab al-jabr wa'l-muqabala (Rules of restoring and equating), which evolved into today's high school mathematics text.

Experimental and theoretical analysis

Experimental:

- Write a program implementing the algorithm
- Run the program with inputs of varying size and composition

Theoretical:

- Uses a high-level description of the algorithm instead of an implementation
- Characterizes running time as a function of the input size n
- Takes into account all possible inputs, and looks at worst-case
- Allows us to evaluate the speed of an algorithm independent of the hardware/software environment

Why Algorithm Analysis?

- Composability. If an algorithm is proven correct and has a guarantee on its running time, we can reuse it as a subroutine.
- Scaling. Asymptotic running time tells us how the running time scales with the problem size.
- Algorithm Design. Analysis often leads to insights that lead to better algorithms.
- Understanding. Analysis can teach us what parts of algorithms are important for what kind of input, so we can more easily solve related problems.
- Complexity theory. "How hard is problem X?"