Basic Course Information

Data Structures and Algorithms

Data Structures and Algorithm is a basic course on the design and analysis of algorithms. On one hand, the course will focus on presenting general techniques for designing efficient algorithms, such as dynamic programming, divide and conquer, greedy methods, etc. On the other hand, we will study how to compare different algorithms by formally analyzing their complexity (e.g. running time). The problems will will study involve sorting, searching, graph algorithms, computational geometry, and a sprinkling of other topics which vary from year to year. The techiques we look at include, divide and conquer, use of randomization, dynamic programming, approximation algorithms, and using parallelism.

The people

- **Instructor: Michael Goldwasser**
  e-mail: `wass@cs.stanford.edu`
  office: Gates 482
  telephone: 723-4532
  office hours: Tu/Th 3:05–4:30pm (or by appointment – Tu/Th).
- **Teaching Assistant: Iacovos Koumas**
  e-mail: `koumas@cs.stanford.edu`
  office: Gates 193B
  telephone: 723-6077
  office hours: M 4:00–6:00 pm
- **Secretary: Phyllis Winkler**
  e-mail: `winkler@cs.stanford.edu`
  office: Gates 495
  telephone: 723-4377

The lectures

- **Class Lectures: Prof. Michael Goldwasser**
  time: Tuesday/Thursday, 1:15–3:05pm
  place: Skilling 191
  broadcast: Stanford channel E4 (live)

- **Recitation Section: Iacovos Koumas**

time:	Friday, 1:15–2:05pm
place:	Skilling 193
broadcast:	Stanford channel E4 (live)

Prerequisites

The official prerequisite for this course is CS109AB (Introduction to Computer Science) or the equivalent. Although this course will be entirely a "pen-and-paper" course, (i.e. no programming), familiarity with programming concepts such as pointers, arrays, records, procedures, and recurssion will be needed.

In addition, a solid background in mathematics will prove invaluable for this course, in that we will argue formally about the correctness and performance of almost every algorithm and data structure we introduce. The ability to write clear and formal proofs will be relied upon.

Course Mechanics

The primary goal of this course is to teach you how to design and analyze algorithms and data structures. The material will be presented in two weekly lectures. The lectures will be nearly two hours long, and so class participation is encouraged during class. In addition, there will be a recitation section once each week, for which attendance is recommended.

- **Homework (50%)** There will be six weekly homework assignments, on which students are to work alone. For your overall grade, we will throw out the lowest of your six homeworks. They will be handed out on Tuesday in class, and will be due the following Tuesday, at the *beginning* of class. Since it is important to stay up-to-date, and a new homework will go out each Tuesday, *no late homeworks will be accepted.*

  Each homework will consists of some suggested practice problems, and then four or five required problems which you will turn in. Throughout this course, problems will require a formal proof by default. Try to be both clear and precise. The organization of your solutions is just as important as the correctness.

- **Midterm (20%)** In the fourth week, we will give a take-home midterm in place of a homework assignment. Effectively, the midterm has weight equal to a "double-homework." The format for the midterm will be similar to the homeworks, except that it may be more comprehensive.

- **Final (30%)** This final will be in-class, and closed book. It is intended to test both breadth and depth of understanding, and will consists of a variety of both short answer and more involved questions.

- **Extra Credit (5%)** Most of the homeworks, and each of the exams will have an extra credit problem. Your scores on these questions will be kept *independently* of the rest of the scores, and your total extra credit score for the class will add up to 5% on top of the overall grade.

Online Information

Web Page:	`http://www-leland.stanford.edu/class/cs161`
Directory:	`/usr/class/cs161`
	`ftp://ftp.stanford.edu/class/cs161`
Newsgroup:	`su.class.cs161`

We will attempt to keep all online information related to the course up-to-date. Announcements regarding homework problems, common questions, etc will be posted to the newsgroup as necessary. As this is also the place to look for late breaking news, it is important to check this newsgroup regularly.

There will be a class web page, to provide current information about office hours, lecture topics, due dates, and pointers to electronic copies of all handouts.

Electronic copies of all handouts will also be available in the class directory on `leland`, in postscript, dvi, and latex formats. You may access them through a "leland" account if you have one, or through anonymous ftp at `ftp.stanford.edu` in the directory `/class/cs161/`.

Textbook

The required textbook for this class is *Introduction to Algorithms* by Cormen, Leiserson, and Rivest (CLR). Other good references include:

- *Design and Analysis of Computer Algorithms*, Aho and Ullman, Addison-Wesley, 1974.
- *Data Structures and Algorithms*, Aho and Ullman, Addison-Wesley, 1985.
- *The Art of Computer Programming*, Vol. 1–3, Knuth, Addison-Wesley, 1973–.
- *Randomized Algorithms*, Motwani and Raghavan, Cambridge Press, 1995.
- *Algorithms*, Sedgewick, Addison-Wesley, 1988.
- *An Introduction to the Analysis of Algorithms*, Sedgewick and Flajolet, Addison-Wesley, 1996.

Schedule (subject to change)

#	Date	Topic	Reading	Due	Out
1	Tues. June 25	Introduction, Admin., Asymptotics, Sorting	Ch. 1, 2		hw1
2	Thur. June 27	Recursion, Divide & Conquer, Master Theorem	Ch. 3, 4		
3	Tues. July 2	Lower Bound for Sorting, Linear Sorts, Order Statistics	Ch. 9, 10.1, 10.3	hw1	hw2
	Thur. July 4				
4	Tues. July 9	Priority Queues, Heaps, Dynamic Sets, Search Trees	Ch. 7, 13.1–13.3, and pp. 197–198	hw2	hw3
5	Thur. July 11	Balanced Trees	Ch. 14		
6	Tues. July 16	Augmented Data Structures	Ch. 15	hw3	Mid
7	Thur. July 18	Randomization, Quicksort, Median Finding, Treaps	Ch. 6, 8, 10.2, 13.4		
8	Tues. July 23	Skip Lists and Hashing	Ch. 12 and additional reading	Mid	hw4
9	Thur. July 25	Dynamic Programming	Ch. 16		
10	Tues. July 30	Graphs (1 of 3) The basics	Ch. 5.4, 23	hw4	hw5
11	Thur. Aug. 1	Graphs (2 of 3) Spanning Trees	Ch. 24		
12	Tues. Aug. 6	Graphs (3 of 3) Shortest Paths	Ch. 25	hw5	hw6
13	Thur. Aug. 8	Computational Geometry	Ch. 35		
14	Tues. Aug. 13	Advanced Topics (dead week)	parts of Ch. 30, 37	hw6	
15	Thur. Aug. 15	Review (dead week)			
F	Saturday Aug. 17	Final Exam (12:15–3:15)		Final	Final