Course Syllabus: CS 141: Intermediate Data Structures and Algorithms
Winter, 2009

Course Description: Basic algorithm analysis techniques; asymptotic complexity; big-O and big-Omega notations; induction and recurrence relation; efficient algorithms for discrete structures including trees, strings, and graphs; efficient implementation of algorithms; general algorithm design techniques including divide-and-conquer, the greedy method, and dynamic programming; lower bound for comparison based sorting.

Course format: The class consists of two 120-minute lectures and a three-hour lab per week. Labs start from the 1st full week (*i.e.*, Jan. 5) of the quarter. Each lab will be supervised by a TA and will be used primarily to enforce concepts and techniques learned in class and complete the programming assignments.

Prerequisite: CS 14 and CS/Math 111. The students are expected to have a strong background in programming (control structure, subroutine, recursion, pointer, coding in procedural or objected oriented language, style, and debugging) and be familiar with basic data structures (array, linked list, tree, binary tree, stack, queue, etc.) and fundamentals of discrete mathematics (summation, induction, set, series, recurrence relation, combinatorics, graph, discrete probability, etc.)

Instructor: Tao Jiang, EBII 336, phone: 827-2991, email: `jiang@cs.ucr.edu`. Office Hours: Mon. 5-6pm and Wed. 2-3pm.

Teaching Assistants: All office hours are held in EBII Room 110.

Lab 021	M 11:10am - 2pm	EBII 226	Monik Khare	mkhare@cs	TBA
Lab 022	M 2:10pm - 5pm	EBII 226	Sima Lotfi	lotfis@cs	Th 11-12pm

Textbook: A. Levitin (Le). *Introduction to the Design and Analysis of Algorithms*, 2nd edition, 2007, Addison Wesley.

Lecture Notes: Copies of the slides used in lectures are available on the class homepage
`www.cs.ucr.edu/~jiang/141-homepage.html`

Reference Books (on reserve in the science library):
(1) T. Cormen, C. Leiserson, R. Rivest, and C. Stein (CLRS), *Introduction to Algorithms*, 2nd ed., 2001, MIT Press.
(2) M.T. Goodrich and R. Tamassia (GT), *Algorithm Design*, 2002, Wiley;
(3) M. Weiss, *Data Structures and Algorithm Analysis in C++*, 1999, Addison Wesley;
(4) S. Dasgupta, C. Papadimitriou, and U. Vazirani, *Algorithms*, 2008, McGraw-Hill.

Grading:

3 homework assignments (paper work) — 15%
2 (or 3) programming assignments — 15%
Lab performance (attendence and in-lab exercises) — 5%
Midterm test (in class, Feb. 10, Tuesday) — 20%
Final examination (March 16, 11:30am - 2:30pm) — 45%

Reading assignment: You are expected to review, before and after each class, the material to be covered in the class. A reference to the chapters of the text and major reference books that will be covered in lectures can be found in the *tentative timetable* below.

Assignment Policy:

1. All assignments will be handed out in class and/or posted on the class homepage
   `www.cs.ucr.edu/~jiang/141-homepage.html`

2. You have two weeks for each (homework and programming) assignment.

3. Submit your homework assignments to Prof. Jiang at least 5 minutes before the lecture on the due date begins. Make sure to put the assignment in the pile assigned to your lab, and always include your full name, UCR ID, *lab section number*, and the assignment number.

4. Write legibly. What cannot be read will not be graded.

5. Submit your programming assignments electronically. Detailed instructions on the electronic submission procedure as well as deadlines will be explained in the lab and also available on the class homepage.

   Programs will be graded based on correctness as well as commenting, style, readability, and code efficiency. You must include a few pieces of information at the beginning of each program: your name, UCR ID, login, lab section number, and assignment name (as1, as2, etc). Points will be marked off if you do not include this information. You are expected to know good programming style. Your programs are expected to be done using the Linux g++ compiler, they should be divided into multiple files, and you should always write and turn in a makefile. Besides program codes, you will also be required to turn in testing documentations which will be part of your grade.

6. No late assignments will be accepted.

Academic dishonesty: Many students find it helpful to consult their peers while doing assignments. This practice is legitimate and to be expected. However, it is not acceptable practice to pool thoughts and produce common answers. To avoid this situation, it is suggested that students not write anything down during such talks, but keep mental notes for later development of their own. Major occurrences of academic dishonesty, such as the submission of work that is not the student's own, will be dealt with according to the UCR's and CSE's policies on academic dishonesty that can be found at webpage `http://www.cs.ucr.edu/`. Students who allow their files or assignments to be copied are as guilty of academic dishonesty as those who copy and will be treated accordingly. Each student is responsible for taking reasonable precautions to ensure that his/her work is not available for unauthorized use. If you copy solutions from books or the Internet, you need give credit to the authors of the solutions and you will lose credits accordingly.

Table 1: Tentative Timetable

Week of	Topic	Chapters of	
		[Le]	[CLRS]
Jan. 5	introduction to algorithms, fundamental data structures time complexity, growth function, asymptotic notation	1	1,2,3,10
Jan. 12$*$	analysis of nonrecursive and recursive algorithms recurrence relations	2,A,B	3,4,A
Jan. 19$+$	brute force algorithms, divide-and-conquer algorithms	3,4	2,7
Jan. 26$*$	more divide-and-conquer examples	4	28,33
Feb. 2	decrease-and-conquer algorithms, DFS, BFS	5	22
Feb. 9$+$	**midterm** the transform-and-conquer method	6	6,12,13,28
Feb. 16	dynamic programming algorithms	8	15, 25
Feb. 23$*$	greedy algorithms	9	16, 23, 24
Mar. 2	space and time tradeoffs, string matching	7	6
Mar. 9	lower bound for sorting, review	11	4

Legend: $*$ and $+$ denote the handing out of homeworks (on Mondays) and programming assignments, respectively.