CS 430 – Introduction to Algorithms
Preliminary Syllabus (version 1.1)

Gruia Calinescu
Department of Computer Science
Illinois Institute of Technology

Fall 2009

1 Course Summary

Welcome to CS 430 Introduction to Algorithms. In this course we study basic techniques for algorithm design. We also use basic analysis methodology of the complexity of algorithms, with worst case and average case bounds on time and space usage. For this, we use the "Big Oh", "Theta", and "Omega" notation.

We start with data structures, and their application for efficient algorithmic solutions. Then we concentrate on efficient sorting techniques, followed by general techniques such as Greedy, Divide-and-Conquer, and Dynamic Programming. We continue with basic graph algorithms, followed by specific examples from string matching and computational geometry.

We will also discuss a practical case study, including identifying the situations when the theory of algorithms is necessary, and the search for an appropriate algorithm.

Finally, we look at the notion of NP-Completeness as a defining characteristic of the hardness of a problem.

All Main Campus students, including those registered in the Internet section, are expected to attend a recitation session or lab Fridays 11:25-12:15.

2 Textbook

The required textbook is *Introduction to Algorithms* (second edition) by Cormen, Leiserson, Rivest, and Stein, McGraw Hill, 2001. It should be available at the bookstore and on the web, where `www.addall.com` might help you find the best prices. Our library has on-line access to this book: from `http://http://www.gl.iit.edu/database/database.htm` look for Books 24x7, login and search for "Algorithms".

3 Prerequisites

CS 330 and CS 331, or CS 401. To be precise, I assume familiarity with:

1. Elementary data structures: stacks, queues, arrays, linked lists and trees.

2. Discrete structures: sets, trees, directed and undirected graphs.

3. From calculus: functions, polynomials, matrices, and logarithms.

4 Getting Help

Office hours are Monday 2:00 - 3:00 PM and Wednesday 3:00 - 4:00 PM, in room SB 229B, or by appointment. I will also be available up to 15 minutes after the class, in the class. For an appointment send email to `calinescu@iit.edu`. You can also call me at 312-567-5273.

The TA for this class is Rodney Summerscales, who will guide you during the **recitation** hours Fridays 11:25-12:15. Rodney (rsummers@iit.edu) has office hours Fridays from 12:30-2:30 PM in SB 006, 312-567-5339.

Some handouts (including this syllabus, homeworks, and some solutions) will be available at `http:// www.cs.iit.edu/~cs430`. Students are expected to read their email (the "official" email address we have on their file) every weekday. Announcements will also be posted on the web page.

5 Grading

The grading allocation is given below.

Homeworks	42%
Project(s)	20%
Quizzes	8%
Midterm	10%
Final exam	20%

Students in the TV or Internet sections will have the weight of homeworks increased up to 50% and the weight of quizzes decreased to maybe 0%, depending on where they are located and which days we'll hold the quizzes. All are encouraged to submit answers to the quizzes; we will grade those and give useful feedback.

The midterm and the final exams are closed books and closed notes, and will contain, among other problems, modified homework problems. The midterm will be on October 28, in class. The final exam is not scheduled yet, but will be according to the IIT calendar in the period December 8-14. Plan for the worst case - an exam on 14th.

There will be circa eight (possibly, short unannounced) quizzes, during either class or recitation. Six homeworks will be assigned, and are to be solved individually. Seek help from me if you are having any difficulties with the homework. Many homeworks and exam problems will ask for designing an algorithm to solve a typical problem. Unless specified otherwise in the description of the problem, the problem includes arguing correctness, and giving and justifying good upper bounds on the running time of the algorithm. Each of these two reasonings (for correctness and running time) will be worth 10% of the grade on that particular problem.

Copying homeworks and exams will be penalized, from getting no points on the specific problem (first occurrence) to getting no points on the whole assignment and being reported to the the Chair of the Computer Science Department and the Dean of the Undergraduate College. A failing grade and removal from the University might be the ultimate consequence of cheating. The project(s) will probably be a team effort, and guidelines regarding use of other resources will be included in the description of the project.

The penalty for late assignments is: 10% one lecture late and 20% one week late. No assignment will be accepted if more than one week late.

A composite score will be computed according to the scale above. This score will be adjusted such that your final score percentage is at most 10% more than your composite exam score. For example, with 67% on exams, and 91% on homework, project, quizzes, and class participation, the final score will be 77%, which is lower than the weighted average. The official grade will follow the following scale, which I might adjust by + or - 2% to ensure consistency with Spring 2002 and Spring 2006 (when there were 10 A's, 22 B's 21 C's 11 D's and 4 F's). Be aware that graduate students cannot get a D and instead will get an E. There is no prescribed curve and you are not competing with each other.

90 - 100%	A
80 - 89 %	B
70 - 79%	C
60 - 69 %	D
0 - 59 %	E

Reasonable accommodations will be made for students with documented disabilities. In order to receive accommodations, students must obtain a letter of accommodation from the Center for Disability Resources and make an appointment to speak with me [the instructor] as soon as possible. The Center for Disability Resources (CDR) is located in Life Sciences Room 218, telephone 312-567-5744 or `disabilities@iit.edu`.

6 Topics to be covered

1. Algorithm Analysis and Mathematical Background (Chapters 2, 3, 4)
2. Heaps, Heapsort (Chapter 6)
3. Divide-and-Conquer method and Quicksort, Medians (Chapters 7, 9.2)
4. Hash Tables (Chapter 11)
5. Binary Search and Red-Black Trees (Chapters 12, 13)
6. Data Structure for Disjoint Sets (Chapter 21)
7. The Greedy Method (Chapter 16)
8. Dynamic Programming (Chapter 15)
9. Elementary Graph Algorithms (Chapter 22)
10. Minimum Spanning Trees (Chapter 23)
11. Shortest Paths (Chapters 24 and 25)
12. Pattern Matching (Chapter 32, time permitting)
13. Convex Hulls (Chapter 33, time permitting)
14. NP-Completeness (Chapter 34)