CS 364 – Winter 2002
Location: SS 2117
Time: W, 7-9 pm, Tutorial W 6-7

Instructor: Toniann Pitassi
Office: 2305A Sandford Fleming Building
email: toni@cs.toronto.edu
Office hours: W 3 - 4 pm

Course Web Site: http://www.cs.toronto.edu/ toni/Courses/364-2002/CS364.html
Refer to this site periodically for important announcements and other information. All handouts will be available on the site in postscript form.

Course Notes: Posted on course web site.

Text: There is one required text for this course. "Introduction to the Theory of Computation" by Michael Sipser, PWS Publishing Company. The most relevant chapters for this course are 0,3,4,5,7, and 8.

Course Description

This is a first course in the theory of computation and the theory of computational complexity. In part I of this course, we will address the following question: What problems can and cannot be solved by a computer program? We will formulate the concept of a computer program mathematically, and prove many interesting results. In particular, we will show that many problems have no computer solutions. After understanding which problems are and are not computable, we will restrict our attention to those problems that do have computer program solutions. Now we are interested in the most efficient algorithm that can solve the given problem. In part II of the course, we will define polynomial-time computatioins, and we will introduce the famous complexity classes P and NP, and discuss which problems are in each of these classes, and techniques used to demonstrate the hardness of a particular problem. Finally in part III of the course, we introduce two important algorithmic techniques that are often used to give polynomial-time algorithms for well-studied problems. These techniques are: (i) the greedy method, and (ii) dynamic programming. Below we give a more detailed list of the topics that will be covered.

(1) (5 weeks) **Turing Machines and Computability** Using the Turing machine model, we can characterize precisely which languages can be recognized by a Turing machine (and hence by any computer). Then we show the famous result that there are many problems that cannot be solved by any modern day computer.

   (a) Preliminaries (Sipser, Chapter 0)

   (b) Turing machines, and the Church-Turing Thesis. (Sipser, Chapter 3)

   (c) Decidability (Sipser, Chapter 4)

   (d) Reducibility (Sipser, Chapter 5)

(2) (5 weeks) **Polynomial-time computation and NP** Complexity theory is about rigorously defining what it means for problems to be hard or easy to compute, and then trying to understand in mathematical terms which problem are inherently hard versus which ones are easy.

   (a) Time complexity and the complexity classes P and NP. (Sipser Chapter 7.1-7.3)

   (b) NP-completeness (Sipser Chapter 7.4-7.5)

   (c) Space Complexity (Sipser, Chapter 8), time-permitting

(3) (3 weeks) Polynomial-time algorithms and techniques for efficient algorithm design: the greedy method, and dynamic programming. For this part of the course, we will use lecture notes.

Marking Scheme:

3 assignments worth 5% each (Due: February 6, March 13, April 10 respectively.) 2 closed book tests worth 22.5% each. (February 13, March 27) Final exam worth 40%

Assignments are due at the *beginning* of tutorial, since solutions will be discussed during the tutorial. In marking assignments, the style of the written presentation will be a significant portion of the grade.

The work you submit must be your own. You may discuss problems with each other; however you should prepare written solutions alone. You are not permitted to consult other texts, papers or web sites for solutions. Copying assignments is a serious academic offense and will dealth with accordingly.

This is a theoretical course. Certain concepts may be difficult to understand at first and will require diligence. Practicing on many problems (homework assignments and problems at the end of each chapter) is often the most effective way to learn the course material. You are also encouraged to ask many questions in class, during tutorial, and during office hours.