CSC364H – Summer 1999

Instructor:	Michael Soltys, e-mail: `soltys@cs.toronto.edu`
Lectures:	MW1 in SS2102 (L0101)
	T7-9 in SS2102 (L5101)
Office Hours:	Tuesdays 5:10-6:00 in UC48
	Wednesdays 12:10-1:00 in UC48

Tutorials: There are two sections. Students from Victoria, Trinity, Univ College, St. Mikes and Innis should go to section 1, and students from New College, Woodsworth, Erindale and Scarborough, should go to section 2.

L0101 W2	section 1:	SS2102, Steve Myers
	section 2:	SS2106, Kit Ng
L5101 T6	section 1:	SS2102 Mohammad Mahdian
	section 2:	BL113 Alan Skelley

Course web page: `http://www.cs.toronto.edu/~soltys/csc364.html`
Check this web page regularly for announcements and solutions.

Text: "CSC364H Summer 1999 Readings", available from UofT Bookstore. It contains parts of the first five references below.

References:
Cormen, Leiserson and Rivest: "Introduction to Algorithms"
Brassard and Bratley: "Fundamentals of Algorithmics"
Motwani and Raghavan: "Randomized Algorithms"
Kfoury, Moll and Arbib: "A Programming Approach to Computability"
Sipser: "Introduction to the Theory of Computation"
Garey and Johnson: "Computers and Intractability"

Marking Scheme:
3 Assignments, worth 5% each
3 Closed book quizzes, worth 15% each
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 proportion of the grade. Each assignment is followed by a quiz in tutorial, one week later.

The work that you submit must be your own. You may discuss problems with each other; however, you should take no notes during these discussions, and you should prepare written solutions alone. Copying assignments is a serious academic offense and will be dealt with accordingly.

Course Outline:

1. Polynomial time algorithms and techniques for efficient algorithm design: The greedy method, dynamic programming, and randomized algorithms. (5 weeks)

2. The theory of NP completeness: The classes P and NP, polynomial time reducibility, intractable problems. (5 weeks)

3. Computability Theory: Simple models of computation, Church's thesis, computable and non-computable functions, universal programs, undecidable problems, recursive and recursively enumerable sets. (3 weeks)