Computer Science 3719 – Winter 2011, MUN
Theory of Computation and Algorithms

Basic course info:
Instructor: Antonina Kolokolova.
Email: kol@cs.mun.ca. *Office:* ER 6033
Lectures: Tue/Thu 12:00-12:50 and Fri 13:00-13:50 in EN 1051
Instructor office hours: TBA.
Course prerequisites: CS 2711 and PM 2320. If you have taken CS 2742, and are taking PM 2320 concurrently, this might be sufficient, but you should be willing to learn extra material on the fly, if necessary.

Course Web Site: `http://www.cs.mun.ca/~kol/courses/3719-w11`
All announcements will be posted on the website, as well as answers to frequently asked questions, assignments, solutions and so on. Please check it regularly.

Textbook:
There will be no main textbook for this course; some course notes will be posted online. There will be three **reference books**.There will be three reference books. The main reference book will be Sipser: intro to the theory of computation.

1. M. Sipser: *Introduction to the Theory of Computation* (2nd edition).
2. Cormen, Leiserson, Rivest, and Stein: *Introduction to Algorithms* (3rd Edition).
3. Kleinberg and Tardos: *Algorithm design.*

Marking scheme (tentative!):
4 assignments of 10% each, one midterm test 20% and a final exam 40%. Note that the last assignment may be due during the last week of the semester (to provide an adequate preparation for the final exam).

Course Outline:

What is an algorithm? What does it mean for a problem to be computationally easy, hard or unsolvable? What can be solved by a computer with only small finite memory (or no memory)? This course is an introduction to the theory of computation, an area which studies these types of questions. We will talk about what is known (and what is open) about the power of computation.

We will start with a simple memoryless model, finite automata, and show that they can compute precisely everything expressible by regular expression. Then we will talk about more powerful kinds of automata; then the Turing machine: it will allow us to say what is an algorithm. We will move on to complexity theory; in particular, we will study problems for which no algorithm better than a brute-force search for a solution is known (or believed) to exist (the famous million-dollar P vs. NP problem). We will then show that for some problems no algorithm exists at all, no matter how inefficient.