Algorithms and Data Structures
COMP 251

McGill University, Winter 2013

Course Details

Time: Monday, Wednesday 13:05–14:25
Place: ENGTR 0100

Instructor: Professor Clark Verbrugge
Office: McConnell, room 230
Office hours: Monday & Wednesday 14:30–15:30, Friday 11:00–12:00, or by appointment.
Phone: 514 398-2411
Email: `clump@cs.mcgill.ca`

Teaching Assistants: TBA

Email, Website

McGill's MyCourses will be used for course announcements, to manage assignments and for online discussions. Students are expected to monitor their McGill email account for course-related news and information.
The external course website is: `http://www.sable.mcgill.ca/~clump/comp251`

Pre-requisites

- COMP 250 (Introduction to Computer Science) or COMP 203.

Note: MATH 240 (Discrete Structures 1) is not required but is recommended.

Students registering without the pre-requisite may find the course removed from their transcript by their Faculty. Please consult the instructor if you do not have all the pre-requisites.
Not open to students who have taken or are taking COMP 252.

Textbook

Reference material is available in the following texts:

- Jon Kleinberg and Éva Tardos. *Algorithm Design.*
  This is the main, required text for the course. It is available at the bookstore and on reserve in the Schülich library.

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. *Introduction to Algorithms,* (any edition).
  This is supplemental text is a more comprehensive reference that includes significant material on data structures. It is also available electronically for free:
  `http://library.books24x7.com/toc.asp?bookid=3444`

Evaluation

5 Assignments:	30%
Midterm:	20%
Exam:	50%

A supplemental exam (70%) will be held if required.
In accord with McGill University's Charter of Students' Rights, students in this course have the right to submit in English or in French any written work that is to be graded.

Assignment and Exam Policy: Assignments must be submitted on time. Late assignments will only be accepted in highly-exceptional circumstances, typically requiring a medical note as well explicit permission from the instructor. Note that I do not consider your workload in other courses exceptional, no matter what courses you take! No assignment submissions will be accepted after marked assignments have been returned, or after solutions have been discussed.
McGill University values academic integrity. Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offenses under the Code of Student Conduct and Disciplinary Procedures (see `http://www.mcgill.ca/integrity/` for more information).
Keep in mind that you are taking this course to learn about fundamental algorithms and to gain practice in analyzing and implementing them. It is not a course on using the internet to find solutions: in all cases, to be accepted **work submitted for this course must represent your own efforts.** Copying assignments or tests, or allowing others to copy your work, will not be tolerated. Note that introducing changes into a copied program or assignment is still considered plagiarism.

Course Content

Note: lecture dates are approximate and may shift, and specific algorithm examples may also change.
Chapter.section readings from Kleinberg & Tardos (KT) and from Cormen, Leiserson, Rivest, and Stein 2nd edition (the one online) (CLRS2) are shown next to topics

				Assignment	
Date	**Topic**	**KT**	**CLRS2**	**Out**	**Due**
Jan 7	Introduction	**1**	**1**		
Jan 9	Algorithm analysis	**2**	**2-3**		
Jan 14	Experimental analysis			A1	
Jan 16	Heaps		**19-20**		
Jan 21	B-Trees		**18**		
Jan 23	Red-black trees		**7, 13**		
Jan 28	Graphs	**3.1-3.2**	**22.1**	A2	A1
Jan 30	Graph algorithms: SCC	**3.5**	**22.5**		
Feb 4	Greedy algorithms: shortest paths	**4.4**	**16.1-16.2, 24.3**		
Feb 6	Greedy algorithms: MST	**4.5**	**23**		
Feb 11	Greedy algorithms: Huffman codes	**4.8**	**16.3**	A3	A2
Feb 13	Lower bounds on sorting		**8**		
Feb 18	Midterm review				
Feb 19	**Midterm (6pm–8pm)**				
Feb 20	Recurrences	**5.1-5.2**	**4.1-4.3**		
Feb 25	Divide and conquer	**5.4**	**7, 33.3**		
Feb 27	TBD				
	Mar 4–8: Reading Week				
Mar 11	Divide and conquer examples	**5.5**			A3
Mar 13	Dynamic programming: Bellman-Ford, Floyd-Warshall	**6.1-6.2, 6.8**	**15, 24.1, 25.2**	A4	
Mar 18	Dynamic programming examples	**6.6-6.7**			
Mar 20	Network flows: Maximum flow	**7.1-7.2, 7.5**	**26.1-26.3**		
Mar 25	Network flow examples	**7.10**			
Mar 27	Hashing.	**13.6**	**11.1-11.4**	A5	A4
	Apr 1: Easter				
Apr 3	Perfect hashing, applications		**11.5, 32.2**		
Apr 8	Concurrency				
Apr 10	Heuristic algorithms				A5
Apr 15	Review				