Intermediate Algorithm Design and Analysis
Computer Science 320

Lecture. MWF 14:00 - 15:00 MWF, DMP 110.

Instructor. Will Evans (will@cs.ubc.ca) 822-0827, ICCS X841.

Tutorials.	T1B	M 16:00 - 17:00	DMP 101	Shiwen He (shiwenhe@cs.ubc.ca)
	T1C	T 13:00 - 14:00	DMP 101	David Buchman (davidbuc@cs.ubc.ca)
	T1D	T 10:00 - 11:00	FSC 1613	Srujan K. Enaganti (esrujan@cs.ubc.ca)

Office Hours.	M 15:00 - 16:00 or by appointment	ICCS X841	Will
	M 17:00 - 18:00	ICCS X150	Shiwen
	W 10:00 - 11:00	ICCS X150	David
	T 11:00 - 12:00	ICCS X150	Srujan

Text. Introduction to Algorithms by Cormen, Leiserson, Rivest, and Stein. It is not absolutely necessary to buy the text, but it is a good idea, especially if you have trouble taking notes.

Webpage. `www.ugrad.cs.ubc.ca/~cs320`

Prerequisites. Ability to write a proof using mathematical notation. Familiarity with summations, sets, relations, functions, asymptotic notation (O, Ω, Θ), recursion, loop invariants, basic data structures (stacks, queues, linked lists, heaps, graphs, hash tables), sorting algorithms, graph traversal (depth-first and breadth-first search).

Formally: Either (a) CPSC 221; or (b) one of CSPC 216,CPSC 260 and one of CPSC 220, EECE 320. In addition, either (a) 6 credits of 2nd year MATH or STAT; or (b) 3 credits of 2nd year MATH or STAT with a grade of at least 72%.

Content. The course is about algorithms. We will discuss how to design an algorithm, how to prove that it is correct, how to analyze its performance, and how to determine if this performance could be improved or not. Algorithms solve computational problems. The problems we will consider come from many areas: statistics, data bases, string processing, bio-informatics, scheduling, networks, and more. Throughout, we will emphasize algorithm design techniques: iterative (input consuming/output producing), recursive, greedy, and dynamic programming.

Introduction. Algorithms. Loop invariants. Running time. Asymptotic notation. Recurrence relations.

Sorting, Selecting, Searching. Randomized algorithms. Lower bounds.

Graphs. Minimum spanning trees. Shortest paths.

Strings. Pattern matching. Sequence alignment.

Data Structures. Amortized analysis. Binomial and Fibonacci Heaps.

NP-completeness.

Course Work. There will be (almost) weekly problem sets, two in-class midterms (October 7 and November 4), and one final exam. Problem sets count 25%, each midterm counts 15%, and the final exam counts 45% toward the final grade.

Problem sets are to be done individually. Please do not accept solutions from or give solutions to other students. Please do not copy solutions from the internet, books, or journals. Feel free to read the internet, books, and journals, but don't plagiarize. See `http://www.cs.ubc.ca/about/policies/collaboration.shtml` for more information.

The penalty for late submission is 20% per day.

If you miss an examination for a medical reason, obtain a note from the Student Health Service or your physician and we will schedule a make-up examination.