CS 385 Algorithms
Spring 2014

Catalog description:
This is a course on design and analysis of algorithms. It covers asymptotic complexity analysis, space-time tradeoffs, standard algorithm design techniques, and classic algorithms that serve as examples of design techniques. This course also develops C++ language programming skills by implementing graph algorithms, numerical algorithms, and complex search and sort algorithms.

Instructor:

Matt Burlick (mburlick@stevens.edu)

Office Hours:	Lieb Building, Room 214	
	W	1:00-3:00pm
	R	2:00-4:00pm
	And by appointment (subject to change)	

Teaching Assistant:

Miaomiao Zhang (mzhang2@stevens.edu)

Meetings:

Lecture:	MRF	1:00pm - 1:50pm	Babbio 204
Recitation:	F	2:00pm – 2:50pm	Babbio 202

Prerequisite:

CS 181 or CS 284

Required textbooks:

- Introduction to the Design and Analysis of Algorithms, 3rd edition, 2012; by Anany Levitin; ISBN-10: 0-13-231681-1; ISBN-13: 978-0-13-231681-1

Policies and Grading

- You, your instructor, and the TA are bound by the Stevens Honor Code. Students are responsible for reading and understanding the course policies in this syllabus and for announcements made in class and in the course email list.

- Since this course involves computer programming, while the majority of your work should be original, if at any point you use part of someone else's solution you MUST cite the source of the code. Copy from others (online or classmates) results in an automatic zero for the assignment and additional possible penalties (including course failure and/or escalation to the honor board).

- During exams, you are **not** permitted to use notes, books, computing or communication devices unless a different policy is specifically announced by the instructor.

- Notebook computers are not to be used in class unless otherwise specified.

- During lecture and recitation sessions please refrain from using mobile phones or otherwise being impolite.

- The course score is a weighted average of the following categories.

Assignments	40%
Midterm	20%
Final	30%
Participation and Attendance	10%

- The course score is on a scale of 100 and letter grades (including plusses and minuses). In addition to the grade break-up provided above, effort and progress may be taken into account when computing your final grade. Final letter grades will be assigned according to class-wide grade clustering.

- Attendance will be taken at both lectures and recitation (attendance is mandatory). You are permitted 1 missed recitation.

- The instructor reserves the right to give a higher grade than your course score, if your performance on later assignments and exams is very strong.

- There are no make-ups for exams. The only possible exceptions are in the case of death in the student's immediate family or near-death experience of the student; advance notice is required.

- Assignments will be accepted up to 48 hours late: up to 24 hours late will get a penalty of 20% grade reduction; up to 48 hours late gets penalty of 50%.

- Except when groups are explicitly allowed, work must be done individually. You are encouraged to discuss the problems with your classmates but you must not share details of the solutions. If you are unsure whether you have shared too much, discuss the situation with the TA or instructor; it is your obligation to avoid even the appearance of cheating.

CS 385 Week by Week

The following is an outline for the course.

	Topic(s)	Reading
Week 1	Course Introduction. Important problem types. Review of fundamental data structures.	Chapter 1
Week 2	Analysis framework; big-O, Theta, Omega. Analysis of non-recursive algorithms.	Sections 2.1-2.3
Week 3	Analysis of recursive algorithms. Master theorem.	Section 2.4, Chapter 5 Intro
Week 4	Brute force algorithms. Exhaustive search.	Sections 3.1-3.2, 3.4
Week 5	Divide and conquer algorithms; Review Mergesort, Quicksort, Binary search.	Sections 5.1-5.2
Week 6	Decrease and conquer algorithms; DFS, BFS, and topological sorting.	Chapter 4, Section 3.5
Week 7	Transform and Conquer	Sections 6.1-6.4
Week 8	Space-time tradeoffs	Section 7.3
Week 9	Dynamic programming	Sections 8.1-8.2, 8.4
Week 10	Greedy techniques	Sections 9.1, 9.3-9.4
Week 11	Iterative algorithms	Sections 10.1-10.2
Week 12	Lower bound arguments; Decision Trees; P, NP, and NP complete problems.	Section 11.3
Week 13	Numerical algorithms. Backtracking, Branch-and-bound.	Sections 12.1-12.2, 12.4
Week 14	Approximation algorithms for NP-hard problems.	Section 12.3
Week 15	TBD	

CS 385 Goals and Assessment

To assess student progress we focus on key skills that can be demonstrated. Here is the official list of course outcomes to be achieved by the end of the semester:

1. Calculate the asymptotic running time of standard algorithms.
2. Explain the meaning of big Oh, Theta, and Omega notations and use them to reason about the performance of diverse algorithms.
3. Use the Master Theorem to prove asymptotic assumptions.
4. Implement standard algorithms using graphs and weighted graphs in C/C++. (e.g. DFS, BFS, MST)
5. Compare and analyze basic and advanced sorting algorithms.
6. Implement advanced search trees such as B tree, AVL tree, and 2-3 tree.
7. Implement numerical algorithms such as Gauss elimination, binary exponentiation, and simplex method.
8. Apply standard algorithm design techniques such as the greedy technique, dynamic programming, hashing, space/time trade-offs, reduction, backtracking, and branch-and-bound.