COMP 6651: Algorithm Design Techniques

Concordia University
Department of CSE

Winter 2016

Instructor: Professor Lata Narayanan
Office: EV 3.163 **Phone:** 848-2424 x3029
email: lata@encs.concordia.ca

Course website: https://users.encs.concordia.ca/~comp6651_4
Please consult the course website regularly for announcements and updates.

Required Text Cormen, Leiserson, Rivest, and Stein, *Introduction to Algorithms*, Third Edition, MIT Press.

Course Description Prerequisites: COMP 5361, COMP 5511.
Mathematical preliminaries; Empirical and theoretical measures of algorithm efficiencies; Optimization techniques and algorithms including greedy algorithms, dynamic programming, and graph algorithms; Amortized analysis; String matching algorithms; NP-complete problems and approximate solutions; Probabilistic algorithms. Term project.

Marking Scheme
The grade will be determined by the following components:
10% Assignments (paper-and-pencil)
10% Programming assignments
10% Problem-solving sessions
10% Quizzes
20% Midterm exam
40% Final Exam

Assignments
Normally, assignments are due 2 weeks after they are handed out. *Late assignments will not be accepted.* All assignments must be uploaded to the Electronic Assignment System (EAS).

Quizzes
There will be a 10-15 minute quiz at the beginning of each class except in Weeks 1, 2, 7, and 13. The quiz will cover material covered in the previous class. Students are responsible to be on time for the quiz. *No extra time can be given to students arriving late.*

Problem-solving sessions
Two group problem-solving sessions will be held during the class in Weeks 2 and 13.

Collaboration policy on assignments
While verbal discussions of assignment problems among students are permitted, all solutions and write-ups must be done independently. This means that you should write your solutions without referring to any written records from any discussions you may have had. Acknowledge your collaborators; you will not lose any points for this. Follow the same principle if you use electronic sources, and be sure to acknowledge such sources. Failure to acknowledge collaborators or sources will be considered a violation of the university's code of conduct.

Academic Code of Conduct
Students should be aware of the university's code of conduct, especially the parts concerning cheating, plagiarism and the possible consequences of violating this code. If you have questions about any aspects of the code, particularly about what exactly constitutes plagiarism, do not hesitate to ask the instructor.

Tentative Schedule

Week 1: Introduction and mathematical preliminaries (Chapters 1, 2, 3, 4)
Week 2: Problem solving session 1 (based on sorting and order statistics (Chapters 6, 7, 8, 9)) Week 3: Quicksort and selection algorithm analysis (Chapters 7 and 9)
Week 4: Greedy algorithms (Chapter 16)
Week 5: Dynamic programming (Chapter 15)
Week 6: Dynamic programming (Chapter 15)
Week 7: Midterm exam
Week 8: Amortized analysis (Chapter 17)
Week 9: Minimum spanning trees, DFS, BFS (Chapters 22, 23)
Week 10: Shortest paths, MaxFlow (Chapters 24, 25, 26)
Week 11: NP-Complete problems (Chapter 34)
Week 12: Approximation algorithms (Chapter 35)
Week 13: Problem solving session 2 (based on material in the entire course)

Important Notes:
You are expected to have taken courses on discrete mathematics and data structures in your undergraduate degree. In particular, it is **highly recommended** that you brush up on the following topics: *sets, relations, functions, logic, proof techniques particularly proofs by induction, graph theory, counting techniques, permutations and combinations, binary search trees, stacks and queues, etc.* It will be very difficult to follow or appreciate this course without such a proper background preparation.