Design and Analysis of Algorithms (CS6161) Syllabus

University of Virginia

Professor Gabriel Robins

Course description (from the graduate catalog): Analyzes concepts in algorithm design, problem solving strategies, proof techniques, complexity analysis, upper and lower bounds, sorting and searching, graph algorithms, geometric algorithms, probabilistic algorithms, intractability and NP-completeness, transformations, and approximation algorithms.

Special emphasis will be placed on **problem solving**, unifying ideas, proof techniques, the "scientific method", as well as striving for elegance, insights, and generalizability in developing algorithms and proofs.

Prerequisites: Discrete mathematics, and undergraduate algorithms course or equivalent

Textbook: "**Introduction to Algorithms**", by Cormen, Leiserson, Rivest, and Stein, Third Edition, 2009

Supplemental reading: "**How to Solve It**", by George Polya, Princeton University Press

Selected papers at: http://www.cs.virginia.edu/~robins/CS_readings.html

Office hours: Tues & Thur 6:15-7:45pm (after every class lecture), and other times by appointment
Also Email Q&A, and a running course-related blog

Class structure: two exams (midterm and final), several problem sets, with problems taken from the textbook and other sources, and a term project (involving implementing and demoing some algorithms and/or related concepts and ideas). Extra credit will be given throughout the semester for solving challenging problems.

We will cover as many selected topics from the following list as time permits:

Fundamentals:
- Review of asymptotics
- Review of basic data structures
- Review of basic algorithms

Sorting and searching:
- Review of classical sorting
- Interpolation Search
- Specialized sorting methods
- Deterministic K^{th} selection
- Lower bounds on max & min
- Majority detection
- Meta algorithms

Advanced data structures:
- Skip lists
- Amortized analysis
- Fibonacci heaps
- Perfect hashing, cuckoo hashing

Graph algorithms:
- Lowest common ancestor
- Minimum spanning trees
- Shortest paths trees
- Radius-cost tradeoffs
- Steiner trees
- Minimum matchings
- Network flows
- Degree-constrained trees

Numerical algorithms:
- Linear programming
- Matrix multiplication
- Karatsuba's algorithm

Distributed algorithms:
- Distributed models
- Asynchronous consensus impossibility
- Leader election in a ring
- Leader election in graphs
- Distributed MSTs

Computational geometry:
• Lower bounds
• Chan's convex hull algorithm
• Segment intersection
• Planar subdivision search
• Voronoi diagrams
• Nearest neighbors
• Geometric minimum spanning trees
• Delaunay triangulations
• Minimum density trees
• Minimum bounding box
• distance between convex polygons
• Smallest Enclosing Circle
• Triangulation of polygons
• Collinear subsets
• Probabilistic analysis

String matching:
• Knuth-Morris-Pratt
• Boyer-Moore
• Edit distance
• Longest increasing subsequence
• Smith-Waterman algorithm

NP-completeness:
• Polynomial time and intractability
• Space and time complexity
• Problem reductions
• NP-completeness of satisfiability
• Independent sets
• Graph colorability
• Travelling salesperson problem
• Approximation heuristics

Grading scheme:

• Attendance	10%	(every student is expected to attend all lectures, except for emergencies)
• Readings	20%	(various readings will be due each week, as explained below)
• Midterm	25%	(most midterm questions will be minor variations from problem sets)
• Final	25%	(most final exam questions will be minor variations from problem sets)
• Project	20%	(implementing and testing a set of algorithms, including a live demo)
• Extra credit	10%	(EC given for solving additional problems, and for more readings)
Total:	110% +	

Weekly readings: The weekly readings in this class consist of a minimum total of 36 items from the recommended readings list at http://www.cs.virginia.edu/~robins/CS_readings.html consisting of various papers, videos, animated demos, Web sites, and books. The required ones are highlighted in red font there, while the rest are "electives". The readings item types should constitute a diverse mix, with a minimum of at least 15 videos, at least 15 papers / Web sites, and at least 6 books. Any items above 36 will count towards extra-credit.

The minimum writeup requirements for these readings are a 2 paragraph description for each paper / video / Web site, and 2 page description for books (longer writeups are of course welcomed also). Each writeup should summarize what you learned and what you found interesting / surprising. At least two submissions are due each Monday (by 5pm, beginning the second week of classes), and more than two submissions per week are of course very welcomed and highly recommended (late penalties will be assessed for late submissions). This policy is designed to help you avoid "cramming" at the end of the semester, and also to help you retain more of the knowledge by pacing it more evenly over time. Please Email all readings submissions to the class Email account at homework.cs6161@gmail.com

Study groups: You are encouraged to work on problem solving and the problem sets in study groups (of size no more than six people). These study group are intended to foster collaborations, encourage brainstorming, create excitement, and make the learning process more fun. Each study group should meet regularly (say twice per week throughout the semester). Everyone in the study group should contribute fairly to the overall group effort (study groups are not meant for people to just copy solutions verbatim from each other; it's OK to share ideas and explanations with each other, and then write your own solutions in your own words, but cutting-and-pasting from other people's work & text is prohibited).

Cheating policy: Cheating and/or plagiarism is strictly prohibited, including under the UVa Honor Code (see http://honor.virginia.edu/ for more details). Violators will be subject to serious penalties, including receiving a failing grade on an exam or even in the entire course, as well as possible referral to the UVa Honor Committee (which could lead to expulsion from the University).

Some examples of cheating / plagiarism include:

- Mass-copying of solutions from other people or sources (including downloading from Web sites);
- Mass-sharing of solutions with other people (including uploading to Web sites);
- Cutting-and-pasting from other people's work and/or text;
- Copying article/book/video/movie reviews from anyone (including from Web sites);
- Having other people solve entire problems for you with little effort on your part;
- Providing other people with verbatim solutions to problems with little effort on their part;

The list above contains only some selected examples of bad behavior and is definitely not exhaustive. It isn't feasible to enumerate all the possible ways to cheat, so please don't look for "loop holes" in this policy, and instead please respect its spirit and positive intent. If you're in doubt as to whether any particular behavior violates the cheating policy or Honor Code, please ask any of the TAs or the Professor. We would like to encourage honest collaborations, brainstorming, and study groups, and we prefer to not play "gotcha" with well-meaning people. We sincerely ask for your help in creating a more positive learning experience for everyone.

Important note: We do have access to automated cheating/plagiarism detection tools (e.g., software that automatically compares the text of submissions pairwise for textual similarity). These tools also employ automated Google searches that systematically compare phrases and sentences from a submission against general Web searches. We also perform manual comparisons across submissions to detect cheating/plagiarism. So please be aware that cheating/plagiarism is easy to detect, and it is definitely not a risk worth taking.