

# Design and Analysis of Algorithms (CS6161) Syllabus

University of Virginia

Professor Gabriel Robins

**Course description:** 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 materials at: [http://www.cs.virginia.edu/~robins/CS\\_readings.html](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

**Office hours:** right after every class lecture (Tues & Thur beginning at 6:15), and other times by appointment (also Email Q&A and course blog); there are also several TAs to help students, with office hours almost every single day (including weekends). The office hours schedule is posted on the course Web site, and we encourage students to meet with the TAs often.

**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:

- Classical sorting algorithms
- Interpolation Search
- Specialized sorting methods
- Deterministic  $K^{\text{th}}$  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 rings and graphs
- Distributed MSTs

**Computational geometry:**

- Lower bounds
- Classical convex hull algorithms
- 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)
• Midterm	30%	(most midterm questions will be minor variations from problem sets)
• Final	30%	(most final exam questions will be minor variations from problem sets)
• Project	30%	(implementing and testing a set of algorithms, including a live demo)
• <u>Extra credit</u>	<u>10%</u>	(EC given for solving additional problems, and for readings)
Total:	110% +	

**Extra-credit readings:** The extra-credit readings in this class consist of items from the [recommended readings list](http://www.cs.virginia.edu/~robins/CS_readings.html) at [http://www.cs.virginia.edu/~robins/CS\\_readings.html](http://www.cs.virginia.edu/~robins/CS_readings.html) which includes various papers, videos, animated demos, Web sites, and books. The more significant ones are highlighted in **red font** there, so please do those first. Any of these readings 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 most two submissions per day are allowed; this policy is designed to help you avoid procrastination and "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](mailto:homework.cs6161@gmail.com)

**Study groups:** You are encouraged to work on the problem sets and on the homeworks in study groups (of size no more than six people). These study groups 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, which is disallowed; it's OK to share ideas and explanations with each other, and then write your own solutions in your own words, but copying-and-pasting from other people's work & text is prohibited. Please form your study groups early in the semester (by the second week of classes) and meet regularly.

**Cheating policy:** Cheating and/or plagiarism is strictly prohibited, including under the [UVa Honor Code](http://honor.virginia.edu/) (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:

- Copying of solutions from other people or sources (including downloading from Web sites);
- Verbatim sharing of solutions with other people (including uploading to Web sites);
- Cutting-and-pasting from other people's work and/or text;
- Posting to Web sites or to blogs course questions / answers (problem sets, exams, homeworks, etc.)
- 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;
- Providing solutions on an exam or assignment that you cannot explain in person;

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 exploit "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 reserve the right to ask you in person any of your answers on any exam / homework / submission (and we will do this at our discretion). **If you can't clearly explain in person one of your answers, we will consider this to be a strong indicator of cheating.** Similarly, if we find one of your answers on the Web verbatim (or near-verbatim), we will consider this to be cheating also.

Remember that this is a course in **algorithms** and **problem solving**, and not a course in "Google searching", nor a course in "copying-and-pasting", nor is it a course in "TED talks binge-watching" (and we will repeat this ethos in class numerous times throughout the semester). The readings and extra-credit opportunities in this course are not a substitute for learning about algorithms, but rather they are designed to ignite your intellectual curiosity, expose you a lot of interesting and cool ideas, help you become a better problem-solver, and elevate the grades of those who diligently study and learn.

We observed over the years that **"cramming" and procrastination are highly correlated with cheating**, so please pace yourself over the semester and don't postpone studying to right before an exam. Additional damage caused by cheaters include creating unnecessary stress and wasted time for your fellow students and course administrators, and it also demoralizes your professors and TAs over time and forces them to become more suspicious and act more strictly and sternly with students.

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 and plagiarism. So please be aware that cheating / plagiarism are very easy to detect, and it is definitely not a risk worth taking.

We would like to encourage honest collaborations, brainstorming, and study groups, but we prefer to not play "gotcha" with well-meaning people, so **please don't play "gotcha" with us**. We sincerely ask for your help and cooperation in creating a more positive learning experience for everyone.