Welcome to CIS 201, Tools for the Analysis of Algorithms. The class meets TuTh from 2-3:45 in Physical Sciences room 136. My office is E2 room 345B. My E-mail is dph@soe, and my office hours will be after class (TuTh 3:45-4:45). There will be two weekly discussion sections with times and places TBD.

Text: Introduction to Algorithms (3rd edition preferred) by Cormen, Leiserson, and Rivest.

TA: Foaad Khosmood

Course Work: This course will have weekly problem sets (generally due at the start of class on Thursdays), a review exam, a midterm, and a final. In order to pass the class, you must both pass the final and have a passing total score.

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homeworks</td>
<td>25%</td>
</tr>
<tr>
<td>Review Exam</td>
<td>5%</td>
</tr>
<tr>
<td>Midterm</td>
<td>25%</td>
</tr>
<tr>
<td>Final</td>
<td>40%</td>
</tr>
<tr>
<td>Attendance/Instructor discretionary</td>
<td>5%</td>
</tr>
</tbody>
</table>

The final exam will be given in the lecture room at the scheduled exam time: Monday June 7 from 4 to 7 PM. The midterm date will be announced later. In the past somewhere around 65–70% of the possible points have been sufficient to pass, but this varies a little from quarter to quarter depending on the difficulty of questions and the strictness of grading.

Homeworks are to represent the students' individual understanding and any help must be properly acknowledged. Presenting others' work as your own is dishonest, and is called Plagerism. Academic dishonesty on any assignment is grounds for not passing the class. To ensure that any solutions you submit represent your own understanding, you are required to wait at least 1/2 hour after any meeting or discussion with other students before writing up your solutions. Some homework assignments may indicate that they can be done in small groups, and each group should turn in a single set of solutions. All help from outside the group must be acknowledged as above. All members of the group must attempt each problem and fully understand the group's solution. It is inappropriate to simply split up the assigned problems among the group members.

Homework grading will penalize illegible handwriting, please print if in any doubt about your cursive writing. Homeworks are to be done on (preferably lined) 8 1/2 by 11 paper with the problems in order (a new page/side per problem is usually appropriate) and stapled together. Homework assignments will usually be due on Thursdays at the start of lecture. Each student is allowed one late submission up to the following Tuesday with a ≈ 10% grade penalty.

CE Base Requirement As requested by the CE department, all CE students must have their CE Data Structures Base requirement signed off to remain in the course. Turn in a copy of your signed sheet on Thursday April 1 so that I can verify your status. See Graduate Advisor Carol Mullane or CE Graduate Director Katia Obraczka for more information.
Resources: All students are responsible for information posted to the Moodle page and SOE class web page.

Description: This course is a graduate level introduction to the analysis of algorithms. It contains some overlap with the upper-division courses CMPS 101 and CMPS 102, but it gives a deeper treatment putting more emphasis on formal analysis. In particular, average case and amortized analysis are dealt with as well as some advanced data structures and algorithms. It is expected that the student already has significant programming experience and is an expert on the elementary data structures (arrays, lists, stacks, queues, heaps, trees, hash tables, etc.) and their analysis. Other assumed background includes discrete mathematics (sets, functions, graphs, discrete probability, proof by strong induction, elementary combinatorics, etc.), and calculus. An advanced undergraduate course in applied graph theory or algorithms (such as our CE 177 or CMPS102) may be helpful, as is any additional mathematical background.

Syllabus: The following lists the topics I expect to cover. All topics include rigorous analysis of resource requirements.

1. Introduction: algorithms, machine models, proofs, asymptotic notation, divide and conquer, recurrences. (Chapters 1-4)
2. Sorting: heapsort, quicksort, radix sort, information theoretic lower bounds. (Read Chapters 6-8)
3. Order statistics: selection, adversary arguments. (Read Chapter 9)
4. Data structures for sets: hash tables, binary search trees, Red-Black trees. (Read Chapters 10-14)
5. Amortized data structures and the Union-Find problem. (Read Chapters 17, 21)
6. Graphs and searching (Chapter 22)
7. Decision Problems, encodings, reductions, and NP-completeness (Chapter 34)
9. Greedy algorithms, Knapsack, Minimum cost spanning trees, Dijkstra’s algorithm. (Chapters 23, 24)

Other recommended books

- Brassard and Bratley: *Fundamentals of Algorithmics* (more elementary)
- Aho, Hopcroft, Ullman: *The design and Analysis of computer algorithms* (the ancient standard)
- Baase and Van Gelder: *Computer Algorithms*
- Brassard and Bratley: *Algorithms, Theory, and Practice*
- Garey and Johnson: *Computers and Intractability.*

I have requested that most of these (plus the class text) be placed on reserve in the science library.