Overview
This course presents fundamental techniques for designing efficient computer algorithms, proving their correctness, and analyzing their performance. Topics to be covered include graph algorithms, greedy algorithms, divide-and-conquer algorithms, dynamic programming, network flow algorithms, computational intractability, approximation algorithms, randomized algorithms, and quantum algorithms.

Prerequisites
CMSC 351. Students are expected to be familiar with basic programming (loops, pointers, structures, recursion), discrete mathematics (proof by induction, sets, permutations, combinations, probability), calculus (manipulation of logarithms, differentiation, integration), data structures (lists, stacks, queues, trees, graphs, heaps), sorting algorithms (MergeSort, QuickSort, HeapSort), and graph algorithms (minimum spanning trees, Dijkstra’s algorithm). If there is any material that seems unfamiliar, please see the instructor or a teaching assistant as soon as possible to discuss it.

Coordinates
Tuesday/Thursday, 11:00 am–12:15 pm, CSI 2117

Instructor
Andrew Childs (amchilds@umd.edu)
Office hours: Tuesday 12:30–1:30 pm (AVW 3225), Wednesday 3:30–4:30 pm (CSS 3100F), or by appointment

Teaching assistants
<table>
<thead>
<tr>
<th>Email</th>
<th>Office hours (in AVW 4103)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manish Purohit</td>
<td><a href="mailto:manishp@cs.umd.edu">manishp@cs.umd.edu</a>  Monday 10–11 am, Thursday 10–11 am</td>
</tr>
<tr>
<td>Robert Adkins</td>
<td><a href="mailto:radkins2@umd.edu">radkins2@umd.edu</a>  Monday 12:30–1:30 pm, Tuesday 1:30–2:30 pm</td>
</tr>
</tbody>
</table>

Piazza
We will use Piazza for class announcements and online discussion. This is the best way to quickly get help from classmates, the TAs, and the instructor. Rather than emailing questions to the teaching staff, we strongly encourage you to post questions to Piazza at https://piazza.com/umd/spring2016/cmsc451/home.

Texts
Evaluation

Your final grade will be determined as follows:

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignments</td>
<td>25%</td>
</tr>
<tr>
<td>Midterm exam</td>
<td>25%</td>
</tr>
<tr>
<td>Final exam</td>
<td>50%</td>
</tr>
</tbody>
</table>

Assignments

There will be 5–7 homework assignments during the course. Assignments will be made available on the course website and will be due in class, at the beginning of class, on the due date. Solutions will be posted on the course website soon after the due date, so extensions will not be granted, and late assignments will not be accepted. However, the lowest assignment grade will be dropped.

Your answers to the assignment problems should be written neatly and concisely, and you should always aim to present the simplest possible solution. Your assignment grades will be based on both correctness and clarity. Graded assignments will be returned in class, and the grades will be recorded at https://grades.cs.umd.edu.

You are encouraged to discuss homework problems with your peers, with the TA, and with the course instructor. However, your solutions should be based on your own understanding and should be written independently. For each assignment, you must either include a list of students in the class with whom you discussed the problems, or else state that you did not discuss the assignment with your classmates.

Exams

The class will include both a midterm exam and a comprehensive final exam. Both exams will be given in class. The tentative midterm date is March 10, 2016, from 11:00 am–12:15 pm. The final exam date is Thursday, May 12, 2016, from 8:00–10:00 am.

Topics

The following is list of topics is tentative and subject to change:

- Introduction and mathematical background
- Graph algorithms: connectivity, bipartiteness, topological sorting
- Greedy algorithms: shortest paths, minimum spanning trees, scheduling, Huffman coding
- Divide-and-conquer algorithms: counting inversions, geometric algorithms, the fast Fourier transform and convolution
- Dynamic programming: weighted interval scheduling, sequence alignment, shortest paths
- Network flows: Ford-Fulkerson algorithm, applications
- NP-completeness: P and NP, reductions, the Cook-Levin theorem
- Approximation algorithms: set cover, vertex cover, traveling salesman problem
- Randomized algorithms: game tree evaluation, constraint satisfaction, minimum cuts, network routing
- Quantum algorithms: quantum circuits, Deutsch and Simon problems, factoring
Academic accommodations

Any student eligible for and requesting reasonable academic accommodations due to a disability is asked to provide, to the instructor during office hours, a letter of accommodation from the Office of Disability Support Services (DSS) within the first two weeks of the semester.

If you plan to observe any holidays during the semester that are not listed on the university calendar, please provide a list of these dates by the end of the first week of the semester.

As mentioned above, extensions to assignment due dates will not be granted for any reason, so that all students can have timely access to solutions. In the event of a medical emergency that affects your ability to complete an assignment or attend an exam, appropriate accommodations will be made. However, you must make a reasonable attempt to notify the instructor prior to the due date, and you must provide written documentation from the Health Center or an outside health care provider. This documentation must verify dates of treatment and indicate the timeframe that you were unable to meet academic responsibilities. It must also contain the name and phone number of the medical service provider in case verification is needed. No diagnostic information will ever be requested.

Course evaluations

Course evaluations are an important part of evaluating instruction. The Department of Computer Science and its faculty take student feedback seriously. Students can go to http://www.courseevalum.umd.edu toward the end of the semester to complete their evaluations.