CSCI B505/INFO I500: Applied Algorithms

Fall 2017, All sections (Residential/Online)

Instructor:
Funda Ergun
Office: Lindley Hall 430C

Disclaimer This is the first time that the online version of this course is being taught; this is also the first time that this course is being required of various MS programs. As a result, this syllabus is tentative. Any changes to it will be reflected on Canvas, so watch out for announcements. Some of the logistics will be on the main page rather than here; this includes AI and office hour information.

Course Content/Outcomes:
Algorithms are at the heart of any computer-related task. In this course, we will teach how to approach the meta-task of algorithm building, as well as look at individual algorithms. We will use mathematical tools for designing and analyzing our algorithms, and get some simple hands-on coding experience. Many students wonder how this course compares to 503; 505 is broader and less deep, has more programming and less math (even though it is still fairly mathematical). If you’re a non-CS major, or someone who cares more about applications than theory, then you chances are this course is the one you want.

At the completion of this course, you will be able to:

1. Know, use, and, if necessary, modify a range of algorithms and data structures for well known problems.
2. Given a problem, be able to develop an algorithmic solution to it.
3. Be able to analyze the above solution for correctness and efficiency.
4. Given different algorithms, be able to analyze and compare them.
5. Have basic knowledge of complexity, upper and lower bounds.
6. Have basic experience in coding simple algorithms.

Residential vs. Online Sections: This course has both residential and online sections. In terms of topics covered, the two sections will be identical. Programming and written assignments will be the same, even though their logistics will be different. Exams will be different; final grades will be calculated using the same formula but curved separately within sections. We will all be using the same canvas page; online students will be using it more heavily. All students are encouraged to make use of the written and recorded material that is presented online, as well as join online discussions on Canvas and Piazza. Residential students should attend office hours face-to-face, come to classes and lab sections, and turn in paper copies of homework assignments in person. The information in this syllabus pertains to all sections, unless something is specifically designated as belonging to a particular group.
Contacting the Instructor and AIs, discussions: We have multiple ways of maintaining communication in addition to lectures and office hours:

- Keep an eye on announcements at all times.
- For questions/discussions about the course that you don’t mind other people seeing, use Piazza (you can access it from the Canvas page).
- For informal chats/interactions, use the discussion page of Canvas. Please go there and introduce yourselves at the beginning of the semester.
- For communication for the instructor and the AIs’ eyes only (for instance, questions related to your grade), email csci505@indiana.edu. This includes stuff that only the instructor can decide on, but you don’t mind the AIs seeing, such as getting sick and missing an exam, etc. We will do our best to respond within 24 hours; if we haven’t, bug us again.
- For confidential stuff that you don’t want even the AIs to see (e.g., you could not make it to class because you developed a reaction to your depression medication), email the instructor at fergun@indiana.edu.

Logistics for Online Students: Everything will be on Canvas, so please familiarize yourself with it. Piazza is a great place to discuss your Canvas-related (and other) issues with other students, AIs, and the instructor. The course will have lectures presented as videos/slideshows, as well as examples presented similarly. You will not have specific lab sessions, but we will assign a special time that the AIs can help with the programming assignments. The online and residential sections will go in lockstep in terms of assignments and lectures.

Office Hours for Online Students: The instructor and the AIs will hold office hours using the Zoom videoconferencing tool. Please see the main page for the course on Canvas regarding logistics. If you cannot make the office hours, please write to csci505@indiana.edu, and we can make an appointment.

Assignments, Exams for Online Students: You are expected to turn in your assignments and exams online on Canvas. Please get a nice scan of your assignment or exam (if you absolutely have to, you can take a picture, but we’d strongly prefer a scanned document). The exams will be take-home, so they will differ in nature from those given to the residential students. You will be able to see the grading on Canvas.

Textbook: Introduction to Algorithms, Third Edition

Prerequisite Knowledge: (Some of the required knowledge will be assigned as reading; make sure you complete the reading assignments even if you know the topic.) You will need high school level algebra (fluency in logs, powers, modular arithmetic, binary numbers), basic discrete math (graphs, sets, proof techniques, basic counting/probability, arithmetic/geometric series), basic logic (including quantifiers), basic data structures/algorithms knowledge (BFS, DFS, sorting, linked lists, trees, stacks/queues, binary search). If you are missing some of these topics, you can find most of them in the textbook. If you are having difficulty in finding them, please contact CSCI505@indiana.edu.
Grade Distribution:

If you are interested in doing a self-driven project an a topic of your choice, please talk to the instructor. If your topic is approved, you can replace a number of written and/or programming assignments with a freeform project. Otherwise, the grading scheme is as follows.

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labs</td>
<td>15%</td>
</tr>
<tr>
<td>Assignments</td>
<td>20%</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td>25%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>40%</td>
</tr>
</tbody>
</table>

Course Policies:

- **PLEASE** if you are having difficulty with anything, talk to the instructor or the AIs as soon as you can. The earlier we tackle difficulties, the easier it is to fix them.

  **General**
  - All assignments, projects, exams are supposed to be individual work, unless the instructor agrees to a different scheme before the work is started.
  - Programming assignments can be written in C/C++, Java, or Python.
  - Dates and deadlines are strict, no makeups will be given unless the student can show extenuating circumstances. Oversleeping is not one of them.

- **Grading**
  - Late penalty is 10 points per day for two days; afterwards you receive a 0. You will have a week for written assignments and two weeks for programming assignments. Please submit your work early in case you have server problems, as we will use the server’s submission date as the date that you submitted your work.
  - A blank answer will automatically get 10% credit; an incorrect answer might not.
  - You WILL lose points for writing too much, or putting down irrelevant stuff, or re-writing the question for an answer. Be concise.
  - Please track your grades online throughout the semester and report any errors ASAP.

- **Integrity**
  - It is critical that you maintain academic integrity throughout this course. The penalty for a first offense is typically big enough to result in a failing grade for the course; if it is a second offense, the instructor views it as grounds for dismissal from the program. Any offense will be automatically reported to the university, aside from the penalty assigned within the course. Beside obvious forms of cheating, accepting help for a program from another student and/or using any part of their code, using answers/subroutines/etc from the web, presenting others’ work as your own in a report without proper citation, as well as other offenses not listed here, are considered plagiarism. If you are unsure about whether a certain act is allowed, check with the instructor.

- **Attendance and Absences**
  - Attendance is up to the student; however, it is the student’s responsibility to make up for any absences.
**Tentative Course Outline:**
The weekly coverage might change as it depends on the progress of the class. This is a very long list; it is likely that the actual topics will be a smaller subset.

<table>
<thead>
<tr>
<th>Week</th>
<th>Content</th>
</tr>
</thead>
</table>
| **Week 1** | • Introduction, what’s an algorithm, algorithm analysis.  
• Reading assignment: Graphs, BFS/DFS. |
| **Week 2** | • Asymptotic complexity.  
• Reading assignment: Mergesort, deterministic quicksort, binary search. |
| **Week 3** | • Divide-and-conquer, recursion.  
• Reading assignment: Arithmetic and geometric series, induction. |
| **Week 4** | • Solving recurrences.  
• Reading assignment: Substitution method. |
| **Week 5** | • Dynamic programming.  
• Reading assignment: LCS |
| **Week 6** | • Greedy Algorithms  
• Reading assignment: Strassen’s algorithm |
| **Week 7** | • Take a breath, catch up.  
• Reading assignment: Panic. Catch up. Relax. |
| **Week 8** | • Review.  
• Midterm Exam |
| **Week 9** | • Minimum Spanning Tree, Prim’s algorithm  
• Reading assignment: Exam questions at home. |
| **Week 10** | • Shortest paths algorithms  
• Reading assignment: Kruskal’s algorithm |
| **Week 11** | • Maximum Flow  
• Reading assignment: All Pairs Shortest Paths |
| **Week 12** | • Take a breath  
• Reading assignment: Take a deeper breath |
| **Week 13** | • NP Completeness  
• Reading assignment: Reductions |
| **Week 14** | • Randomized algorithms, hashing  
• Reading assignment: Review for Final Exam |