CSCI 382
Algorithms and Problem Solving Paradigms

Fall 2018
Lecture: MWF 1:10-2:00, MC Reynolds 108
Website: `http://ozark.hendrix.edu/~yorgey/382/`
Course text (optional): Algorithm Design by Jon Kleinberg and Éva Tardos
`http://www.aw-bc.com/info/kleinberg/index.html`

Instructor: Brent Yorgey, MC Reynolds 310
Office hours: any time my door is open, or make an appointment at
`http://byorgey.youcanbook.me`
Email: `yorgey@hendrix.edu`

Course Description

Introduction to standard algorithms and algorithm design strategies, with an emphasis on constructing rigorous formal proofs of algorithm correctness and performance, as well as translating fluently between theory and practice. Strategies discussed include brute-force, greedy algorithms, divide and conquer, dynamic programming, amortization, and problem reduction. The course includes an introduction to complexity theory and the complexity classes P and NP. Prerequisites: CSCI 151 and MATH 240.

Evaluation

Evaluation will be based on

- 20%: A class participation grade based on attendance, promptness, and participation in class.
- 30%: Weekly problem sets. The lowest score will be dropped.
- 25%: Two midterm exams.
- 25%: Final exam.

Problem sets

The weekly problem sets, typically due **Friday at 1:10pm**, will collectively be worth 30% of your grade. The lowest problem set grade will be dropped.

Discussion and collaboration on the problem sets is encouraged. However, solutions must be written up individually. Copying another student's writeup, in whole or in part, or directly collaborating on a writeup, will be considered an academic integrity violation. **Insightful discussion with others must**

be cited in your homework solutions. You will not lose points for such citations.

Each student has four late days to spend throughout the semester as they wish. Simply inform me any time *prior* to the due date for an assignment that you wish to use a late day; you may then turn in the assignment up to 24 hours late with no penalty. Multiple late days may be used on the same assignment. There are no partial late days; turning in an assignment 2 hours late or 20 hours late will both use 1 late day.

Unless stated otherwise, assignments **must be turned in on physical paper**. It is up to you whether you hand-write your solutions, or typeset them on a computer and print them. If you typeset them on a computer, I strongly encourage you to use LaTeX; the easiest way to get started is by using a free account on `sharelatex.com`. I will provide a LaTeX version of each problem set which you can use as a starting template for writing up your solutions.

Each homework question will be graded on a 5 point scale, as follows:

- 5: The solution is clear and correct. This solution would easily find a home in a research paper.
- 4: The solution contains a few mistakes, but they are mostly arithmetic or of little significance to the overall argument.
- 3: The solution hits on the main points, but has at least one logical gap.
- 2: The solution contains several logical mistakes, but parts of it are salvageable.
- 1: The solution is just plain wrong.
- 0: No attempt is made at solving the problem.

Exams

There will be two midterm exams and a cumulative final exam. For each exam, you will be given the exam questions at least one week in advance (probably longer for the final exam). You may use any resources in preparing your solutions to the exam—including your notes, textbook, online resources, and each other—with the only exception that I will not answer specific questions about the exam. On the day of the in-class exam, you must come with no notes and write out your solutions on a clean copy of the exam.

Attendance

Prompt lecture attendance is expected and required. Unexcused absences will be reflected in your participation grade. If you must be absent for some reason, please let me **and your learning team** know in advance.

If you are absent from lecture (whether excused or unexcused) it is **your responsibility** to obtain notes from other student(s). Do not come to me and ask "what did I miss?". On the other hand, if after obtaining notes you have specific questions or confusions regarding the topics covered, I would be happy to talk with you.

Grading

This class uses the following non-traditional grading scale, which appropriately reflects the difficulty of the course:

- 80 and above: A
- 70–80: B
- 60–70: C
- 50–60: D
- < 50: F

Learning objectives

By the end of the course, you will be able to:

- Explain, implement, and apply standard algorithmic solutions to common problems.
- Appropriately select and apply standard algorithmic problem-solving and proof techniques such as greedy algorithms, divide & conquer, dynamic programming, network flow, and amortization.
- Use Big-O notation and standard tools such as recurrence relations and the Master Theorem to analyze the asymptotic time and space complexity of computational processes.
- Explain basic definitions and results in complexity theory; prove NP-hardness results by polynomial reduction.
- Construct and reason about appropriate formal mathematical models of a given computational problem.
- Write coherent, logically sound proofs of algorithm correctness and asymptotic complexity, using standard tools such as induction and invariants.
- Move fluently between theory and practice by writing programs to implement theoretical results, and using theory to analyze existing programs.

Disabilities

It is the policy of Hendrix College to accommodate students with disabilities, pursuant to federal and state law. Students should contact Julie Brown in the Office of Academic Success (505.2954; `brownj@hendrix.edu`) to begin the accommodation process. Any student seeking accommodation in relation to a recognized disability should inform the instructor at the beginning of the course.

Academic Integrity

All Hendrix students must abide by the College's Academic Integrity Policy as well as the College's Computer Policy, both of which are outlined in the Student Handbook.

For specific ways the Academic Integrity policy applies in this course, please refer to the Computer Science Academic Integrity Policy.

The short version is that academic integrity violations such as copying code or a writeup from another student or the Internet are **easy to detect** and will be **taken very seriously**. A typical sanction for an academic integrity violation on a problem set would be a grade of zero on the problem set *plus* lowering your final letter grade by one letter.

I am aware that solutions for many of the assigned homework problems are available online. *Do not look at them!* It is very hard to refer to someone else's solution without copying it inappropriately, and if you turn in a partially copied solution that you do not understand, I will probably be able to tell.

If you have any questions about how the Academic Integrity policy applies in a particular situation, please contact me.