CSCI 305: Analysis of Algorithms I

Syllabus, Spring, 2018

Instructor: Geoffrey Matthews, x3797, geoffrey dot matthews at wwu dot edu

Office hours: MTWF 10:00-10:50, CF 469

Lectures: MTWF 12:00-12:50am, CF 227

Webpages:

- `https://wwu.instructure.com`
- `https://github.com/geofmatthews/csci305`

Catalog Description: Introduction to the analysis of algorithms and data structures in a mathematically rigorous fashion. Mathematical fundamentals, counting, discrete probability, asymptotic notation, recurrences, loop invariants. Worst-case, probabilistic and amortized analysis techniques applied to sorting algorithms and classic data structures such as heaps, trees and hash tables. Design techniques such as branch and bound, divide and conquer, will be introduced as will correctness proofs for algorithms.

Course Outcomes: On completion of this course students will demonstrate:

- Basic understanding of the mathematical concepts of asymptotic notation, recurrence relations and loop invariants.
- Basic understanding of worst-case, probabilistic and amortized analysis techniques.
- Thorough understanding of the complexity of sorting algorithms and common data structures, such as heaps, trees and hash tables.
- Thorough understanding of divide and conquer.
- Basic understanding of the formulation and analysis of probabilistic algorithms.
- The ability to analyze and formulate solutions for abstract problems.
- The ability to derive the time and space complexity of basic algorithms.
- The ability to use mathematical reasoning in correctness proofs of algorithms.

Text: *Introduction to Algorithms,* 3rd ed., Cormen, Leiserson, Rivest, and Stein.

Quizzes: Pop quizzes may be handed out at any time. After working on them individually we will solve them together in class. They must be turned in during class for credit. No makeups.

Exams: A midterm and a final, according to the schedule below. The final will be cumulative.

Exams are closed book, with the exception that you may consult two pieces of paper during the exam. You may print or write whatever you wish on these pages.

If you have to miss the midterm exam, notify me as soon as possible. I will handle each case individually.

Homework: Homework must be typeset using LaTeX. The easiest way to get started is using one of the online LaTeX tools, such as `https://www.overleaf.com/`.

Homework must be submitted to canvas by midnight of the due date. Late homework will not be accepted. There will always be sufficient time given to complete the homework in a timely manner if you don't wait to the last night.

Homework may not be graded promptly, but solutions will be posted online shortly after the due date. Only a subset of the submitted problems will be graded.

There may be an assignment due during dead week.

Assessment: Assessment will be based on homework, quizzes, the midterm, and the final. There are no extra credit opportunities in this class. Assignment of ± grades is at the discretion of the instructor.

Homework	Quizzes	Midterm	Final
35%	5%	20%	40%

A	B	C	D	F
100-90	89-80	79-70	69-60	59-0

Schedule:

Week	Text section	Notes	Important dates
April 1–7	1, 2	000–020	
April 8–14	A	030–040	
April 15–21	3,4	050–070	
April 22–28	5,C	080–130	
April 29–May 4	6,7	140–150	
May 6–12			Midterm, Wednesday, May 9
May 13–19	8 (§1,2,3), 11 (§1,2)	160–170	
May 20–26	12 (§1,2,3),	180	
May 27–June 2	13	190	Monday, May 28, Holiday
June 3–9	17 (§1,2,3)	200	
June 10–17			Final Exam, Tuesday, June 12, 8:00am

Academic dishonesty: Academic dishonesty policy and procedure is discussed in the University Catalog, Appendix D. All students should read this section of the catalog. Academic dishonesty consists of misrepresentation by deception or other fraudulent means. In computer science courses this frequently takes the form of copying another's program, either a fellow student's program, or copying one from the web. Due diligence should be exercised in the labs at all times, since both copying and letting someone else copy your program are equally culpable. Do not walk away from your computer in the lab without logging out or locking the screen. Do not print out code and then throw it away in the lab trash cans. Do not share files, even if it is just to "show them something." Describe it in words, or talk to them in person, never share code.

Collaboration: Collaboration with your fellow students is a good way to learn. Feel free to share ideas, solve problems, and discuss your programs with other students. However, collaboration is *not* copying. All code should be original. Remember the **Long Term Memory Rule**: After discussing homework with another student, each of you must destroy all written notes, pictures, files that you shared, erase the board, *etc.*. After that, you must watch a rerun of *the Simpson's*, play a round of ping-pong, go for a walk, or do something else unrelated, for half an hour. Then you can take the knowledge you gained from another student and put it to work, since it is now not copying, but learning. You have made it your own.