Computer Science III

Data Structures & Algorithms
Syllabus

Department of Computer Science & Engineering

University of Nebraska–Lincoln

CSCE 310 – Summer 2021

> "Computer Science is no more about computers than astronomy is about telescopes." –Edsger Dijkstra

> "If you want to be a good programmer you just program every day for two years. If you want to be a world class programmer you can program every day for ten years, or you could program every day for two years and take an algorithms class." —Charles E. Leiserson

1 Course Info

Prerequisites: CSCE 156 (Computer Science II) and CSCE 235 (Discrete Math)

Description: A review of algorithm analysis, asymptotic notation, and solving recurrence relations. Advanced data structures and their associated algorithms, heaps, priority queues, hash tables, trees, binary search trees, and graphs. Algorithmic techniques, divide and conquer, transform and conquer space-time trade-offs, greedy algorithms, dynamic programming, randomization, and distributed algorithms. Introduction to computability and NP-completeness.

Credit Hours: 3

Textbook: The *recommended* text book for this course is *Introduction to the Design and Analysis of Algorithms* (any edition) by Anany Levitin. However, no text book is required as there are plenty of free online Data Structures and Algorithms resources:

- My lecture notes: `cse.unl.edu/~cbourke/ComputerScienceThree.pdf`
- Open DSA: `https://opendatastructures.org/`
- *Algorithms* by Jeff Erickson `http://jeffe.cs.illinois.edu/teaching/algorithms/`

Postrequisites: If you are a Computer Science or Computer Engineering major you will need to receive a C or better in this course to continue in the major.

2 Course Overview

Computer Science is not programming. Rather, Computer Science is the mathematical modeling and study of what computation is–what problems have a computational solution and how efficient that solution can be. Thus, a strong foundation in mathematics is essential to your success as a computer scientist. At the heart of computer science are fundamental, discrete structures which we will study in this course. Specifically, you will learn many of the mathematical definitions, techniques, and ways of thinking that will be useful in Computer Science.

2.1 Topics

- A review of algorithms, algorithm analysis and asymptotics
- Brute Force algorithms, backtracking, generating combinatorial objects
- Divide & conquer techniques, repeated squaring, Karatsuba multiplication, Strassen's matrix multiplication, etc.
- Algorithms for linear systems
- Greedy Algorithms: Huffman coding
- Balanced Trees: Heaps, AVL Trees, 2-3 Trees
- Hash-based data structures
- Graph algorithms: DFS, BFS, MSTs, path finding, shortest path
- Dynamic Programming
- Computation and computability

3 Schedule

See Canvas

4 Course Delivery

For summer sessions this course is delivered online only in a (more-or-less) asynchronously manner.

- Daily lectures will be live streamed via YouTube (https://www.youtube.com/c/ChrisBourkeUNL/live), however the time is yet to be determined.
  - Recordings of the lectures will be available immediately following so you can watch/rewatch at your convenience
  - During the live broadcast, Piazza will be used for questions/answers
- For assignments that allow collaboration, you may use any medium you choose.
  - You can establish your own Zoom rooms to talk back and forth and share a screen
  - You may use discord or slack instead
  - You can (in fact should) be using git to share code (but only use private repos)
  - There are (free) online IDEs that allow you both to type in the same editor at the same time: https://repl.it, https://ide.cs50.io/; a more extensive list: https://gist.github.com/rouzbeh84/4bafc9fe4fe02edf506d11997c4674b0
- Written solutions must be submitted through webhandin as PDFs
- Live office hours will be held online via zoom
- Exams will still be run, but asynchronously as “take home” exams that will be released the day-of. You will have a limited but *flexible* time period to complete them and submit it electronically. No collaboration will be allowed on the exams.

5 Accommodations for Students with Disabilities

The University strives to make all learning experiences as accessible as possible. If you anticipate or experience barriers based on your disability (including mental health, chronic or temporary medical conditions), please let me know immediately so that we can discuss options privately. To establish reasonable accommodations, I may request that you register with Services for Students with Disabilities (SSD). If you are eligible for services and register with their office, make arrangements with me as soon as possible to discuss your accommodations so they can be implemented in a timely manner. SSD contact information: 117 Louise Pound Hall.; 402-472-3787

6 Grading

Grading will be based on assignments (both written and programming portions) as well as two exams.

Category	Number	Points Each	Total
Assignments	4	200	800
Midterm	100	1	100
Final	100	1	100
Total			1,000

6.1 Scale

Final letter grades will be awarded based on the following standard scale. This scale may be adjusted upwards if the instructor deems it necessary based on the final grades only. No scale will be made for individual assignments or exams.

Letter Grade	Percent
A+	≥ 97
A	≥ 93
A-	≥ 90
B+	≥ 87
B	≥ 83
B-	≥ 80
C+	≥ 77
C	≥ 73
C-	≥ 70
D+	≥ 67
D	≥ 63
D-	≥ 60
F	< 60

6.2 Assignments

There will be 4 assignments that will consist of both written exercises as well as *substantial* programming problems. You will be expected to follow all instructions on the assignments. Clarity and legibility are of great importance. If homework is sloppy or unclear, points may be deducted. You are not required to typeset your written solutions; however, it is strongly recommended that you do so using LaTeX, markdown or similar typesetting system. Resources for LaTeXare available on the course web page. Source code and all relevant files for programming portions must be handed in using the CSE web handin program. Each assignment will have a fixed deadline based on CSE's server time. No late assignments will be accepted.

Further, programming solutions will be graded using our online webgrader system. Failure to submit compilable or runnable code may result in a zero. You are expected to do your own substantial testing (and to submit valid, working test cases as well), but it is essential that your submissions work on the webgrader.

6.3 Exams

There will be two exams, both of which will be open-book, open-note, open-computer but you may *not* collaborate with anyone in or outside the class on the solutions.

6.4 Grading Policy

If you have questions about grading or believe that points were deducted unfairly, you must first address the issue with the individual who graded it to see if it can be resolved. Such questions should be made within a reasonable amount of time after the graded assignment has been returned. No further consideration will be given to any assignment a week after it grades have been posted. It is important to emphasize that the goal of grading is consistency. A grade on any given assignment, even if it is low for the entire class, should not matter that much. Rather, students who do comparable work should receive comparable grades (see the subsection on the scale used for this course).

6.5 Late Work Policy

In general, there will be no make-up exams or late work accepted. Exceptions may be made in certain circumstances such as health or emergency, but you must make every effort to get prior permission. Documentation may also be required.

Homework assignments have a strict due date/time as defined by the CSE server's system clock. All program files must be handed in using CSE's webhandin as specified in individual assignment handouts. Programs that are even a few seconds past the due date/time will be considered late and you will be locked out of handing anything in after that time.

6.6 Webgrader Policy

Failure to adhere to the requirements of an assignment in such a manner that makes it impossible to grade your program via the webgrader means that a disproportionate amount of time would be spent evaluating your assignment. For this reason, we will not grade any assignment that does not compile and run through the webgrader.

6.7 Academic Integrity

All homework assignments, programs, and exams must represent your own work unless otherwise stated. No collaboration with fellow students, past or current, is allowed unless otherwise permitted on specific assignments or problems. The Department of Computer Science & Engineering has an Academic Integrity Policy. All students enrolled in any computer science course are bound by this policy. You are expected to read, understand, and follow this policy. Violations will be dealt with on a case by case basis and may result in a failing assignment or a failing grade for the course itself. The most recent version of the Academic Integrity Policy can be found at `http://cse.unl.edu/academic-integrity`

7 Summer Session Policy

As a summer session, the course pace and presentation is accelerated. What would normally be covered over 15 weeks is compressed into less than five. Your success in this course depends on your acceptance of this fact and a commitment to putting in the extra work necessary to understand this material in the time that we do have. This means extensive daily review of materials outside of lecture and a diligent attitude toward completing assignments. As such, no late work will be accepted and no makeup quizzes or exams will be given. The compressed time period and logistics of offering summer courses make it extraordinarily difficult to make such considerations.

In addition, the summer version of this course lacks the same resources that would be available during the regular academic year. In particular, the Student Resource Center is closed and there is no recitation section. This may make getting additional help more difficult and you should make the appropriate adjustments or reconsider taking this course during the regular academic year.

8 Communication & Getting Help

The primary means of communication for this course is Piazza, an online forum system designed for college courses. We have established a Piazza group for this course and you should have received an invitation to join. If you have not, contact the instructor immediately. With Piazza you can ask questions anonymously, remain anonymous to your classmates, or choose to be identified. Using this open forum system the entire class benefits from the instructor and TA responses. In addition, you and other students can also answer each other's questions (again you may choose to remain anonymous or identify yourself to the instructors or everyone). You may still email the instructor or TAs, but more than likely you will be redirected to Piazza for help.

In addition, there are two anonymous suggestion boxes that you may use to voice your concerns about any problems in the course if you do not wish to be identified. My personal box is available at `https://cse.unl.edu/~cbourke/email/`. The department also maintains an anonymous suggestion box available at `https://cse.unl.edu/contact-form`.

8.1 Getting Help

Your success in this course is ultimately your responsibility. Your success in this course depends on how well you utilize the opportunities and resources that we provide. There are numerous outlets for learning the material and getting help in this course:

- Lectures: attend lectures regularly and when you do use the time appropriately. Do not distract yourself with social media or other time wasters. Actively take notes (electronic or hand written). It is well-documented that good note taking directly leads to understanding and retention of concepts.
- Required Reading: do the required reading on a regular basis. The readings provide additional details and depth that you may not necessarily get directly in lecture.
- Piazza: if you have questions ask them on Piazza. It is the best and likely fastest way to get help with your questions. Also, be sure to read other student's posts and questions and feel free to answer yourself!
- Office Hours: the instructor and GTA(s) hold regular office hours throughout the week as posted on the course website. Attend office hours if you have questions or want to review material.
- Don't procrastinate. The biggest reason students fail this course is because they do not give themselves enough opportunities to learn the material. Don't wait to the last minute to start your assignments. Many people wait to the last minute and flood the TAs and SRC, making it difficult to get help as the due date approaches. Don't underestimate how much time your assignment(s) will take and don't wait to the week before hand to get started. Ideally, you should be working on the problems as we are covering them.
- Get help in the *right way*: when you go to the instructor or TA for help, you must demonstrate that you have put forth a good faith effort toward understanding the material. Asking questions that clearly indicate you have failed to read the required material, have not been attending lecture, etc. is *not acceptable*. Don't ask generic questions like "I'm lost, I don't know what I'm doing". Instead, explain what you have tried so far. Explain why you think what you have tried doesn't seem to be working. Then the TA will have an easier time to help you identify misconceptions or problems. This is known as "Rubber Duck Debugging" where in if you try to explain a problem to someone (or, lacking a live person, a rubber duck), then you can usually identify the problem yourself. Or, at the very least, get some insight as to what might be wrong.