CS124 / Data Structures and Algorithms / 2023 Spring

Compiled: 2023-01-19 19:36

Instructor: Clayton Cafiero
Email: cbcafier@uvm.edu
Office: Innovation E309
Office hours: T 08:30–9:40 AM and 3:00–4:00 PM; F 11:00 AM–12:30 PM; or by appointment.

GTA: Avi Chawla
Email: achawla1@uvm.edu
Office: Innovation E434
Office hours: T 12:00–1:00 PM, or by appointment.

UTA help desk: S–Th, 5:00–7:00 PM
Locations: S/T/Th: Votey 207; M/W: Votey 209; exceptions: 15 Feb in Votey 207, 16 Feb in Votey 303

Prerequisites: CS 110 with a grade of C- or better

Class meetings: M/W/F 9:40–10:30 AM, Innovation E204

Welcome to the course!

Overview: Design and implementation of linear structures, trees and graphs; examples of common algorithmic paradigms; theoretical and empirical complexity analysis; sorting, searching, and basic graph algorithms; common ways of storing data and the use cases for each, including stacks, queues, trees, heaps, and hash tables. Knowledge acquired will be applied in a multi-part programming project, and demonstrated in frequent quizzes and final exam.

Learning objectives:

- You will gain knowledge of fundamental data structures and algorithms, and recognize the use cases for each.
- You will gain understanding of the correctness of algorithms and their complexity.
- You will demonstrate knowledge of essential computer science concepts and skills through project programming (AOE CSES 3.3.1).

- You will understand how storing data in hardware affects the complexity of algorithms and determines how best to store the data in memory (AOE CSES 3.4.2).
- You will perform algorithm analysis using asymptotic notation to evaluate best-, average-, and worst-case space and time complexity (AOE CSES 3.7.5).
- You will collect, aggregate, clean, and model data in your project, in which you will analyze and compare different data structures and algorithms (AOE CSES 3.6.1).
- You will demonstrate basic proficiency in C++, including object-oriented design, abstraction, and recursion, by going through the software development cycle in lectures and project programming (AOE CSES 3.7.1-3.7.4, 3.7.6).

Course materials: Textbook: *Essential Algorithms: A Practical Approach to Computer Algorithms* by Rod Stephens, Wiley, 2013. ISBN-13: 978-1118612101. Other materials supplied on Brightspace.

Software: You must have a C++ compiler. If you are using macOS or Linux you probably already have a C++ compiler installed. Procedures for installation of a C++ compiler vary substantially by platform/OS.

You should use an IDE suited to C++ development. CLion is recommended and supported, and is available with a free student license.[1]

Starter / scaffolding code will be released on GitHub, and you will make your submissions by providing a link to your GitHub repository.[2] Accordingly, you will need a reasonably current version of Git installed on your machine. If you are using macOS or Linux you probably already have Git installed. For information on installing Git, see: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git.

Brightspace pilot: This course will make use of our new learning management system (LMS), Brightspace. Brightspace will completely replace Blackboard in fall of 2023. We are participating in UVM's Brightspace pilot program, and thus, we may encounter some rough patches over the course of the semester. Please do not hesitate to contact me if you have any difficulty with Brightspace. Your feedback will help improve the user experience for all students come autumn.

[1] https://www.jetbrains.com/community/education

[2] Git is the most widely used source code management tool in the world. GitHub is a service for hosting Git source code repositories.

Weekly schedule of topics
(tentative and subject to change):

Week	Topics covered
1	Modular arithmetic; recursive functions; intro to C++; output manipulation; file input
2	Pointers, addresses, references; C++ templates; overloaded operators; node class for singly-linked lists
3	Memory: heap and stack; node and stack classes; queues; intro to complexity; asymptotic notation
4	More on complexity and asymptotic notation; loops; trees; traversal and search; binary search trees
5	More on trees; AVL and splay trees; B-trees
6	Binary heap; priority queues; intro to sorting; bubble sort
7	More on sorting: selection, insertion, and merge sort
8	Quicksort, bucket sort, radix, and heap sort
9	Searching; Introduction to hashing; Horner hash; hash collisions; separate chaining
10	Linear probing; rehashing; quadratic probing; double hashing
11	Relations and equivalence relations; dynamic equivalence problem
12	Disjoint sets: weighted union and path compression; introduction to graphs; topological sort; shortest path
13	Dijkstra's algorithm; Bellman-Ford; network flows; max flow/min cut
14	Max flow/min cut continued; Ford-Fulkerson, Edmonds-Karp; minimum spanning trees: Prim and Kruskal
15	Review

Project: Over the course of the semester you will complete your project in five parts. This will assess:

- your ability to use C++ templates, classes, and functions to create and analyze the data structures and algorithms taught in the course;
- your ability to understand how various data structures work and to write an insightful analysis supported by the data you collect; and
- your ability to compare and contrast data structures and algorithms and to identify and justify the best implementation for a given problem.

Each part will involve implementation or modification of various data structures and algorithms, testing, and preparation of a written report. *Work must be your own*. Any code or analysis not authored by yourself or an instructor *must be cited* in the project submission. Project parts (5) are weighted equally. There may be brief, preliminary assignments to serve as checkpoints or scaffolding for your work. Points earned for these submissions will be included as part of your overall project grade.

Part	Topic	
1	Selecting your data set; creating your custom class; loading data from file; creating an array of objects; performing a calculation on the array	Wednesday 1 February 11:59 PM
2	Create queue and stack classes; load your objects into queue and stack; push and pop; analysis of results	Wednesday 15 February 11:59 PM
3	Create BST, AVL and splay trees; perform insert, search and delete operations on each; analysis of results	Wednesday 8 March 11:59 PM
4	Comparison of sorting algorithms (*e.g.*, bubble, selection, merge, and heap); analysis of results	Wednesday 5 April 11:59 PM
5	Creating hash tables, rehashing, comparing collision strategies; analysis of results	Wednesday 26 April 11:59 PM

Late policy / extensions: Each project component has a specific due date / time. You may submit work up to 24 hours after the due date / time, however, late submissions will be penalized 20%. Submissions that are more than 24 hours late will not be accepted unless an extension has been granted. I will consider reasonable requests for extensions when extenuating circumstances arise. (It can't hurt to ask.) However, extensions will not be granted if the request for extension is made within 24 hours of the time an assignment is due, except in the most extraordinary, documented circumstances. So if you wish to request an extension, do so early! If an extension is granted, you must submit your work by the agreed-upon extension date.

Early submission for extra credit: If you submit a project component two or more days early, you will get 5% extra credit. If you submit a project component one day early, you will get 2.5% extra credit.

Complete submissions: When you submit, *be careful not to omit files!* If a submission is found to have omitted files (code or report) you will be penalized 10% and contacted via email requesting that you supply the missing file(s). You will have 24 hours to submit any missing file(s). If you do not submit missing files within that time you will receive a substantially reduced score. Hint: *Check your email regularly!*

Quizzes: There will be 12 quizzes administered in class over the course of the semester. Each quiz will concentrate on the most recent course material. Quizzes will be brief, with multiple choice, true/false, fill-in-the-blank, or other short format questions. The lowest two quiz scores will be dropped.

Final exam: The final exam will be administered on 11 May, 7:30–10:15 AM, in Innovation E204, and will assess:

- your knowledge of the concepts related to the data structures and algorithms, their properties, and how they function;
- your ability to identify the time and space complexity and other properties of algorithms; and
- your understanding of algorithmic complexity as expressed in asymptotic notation.

Grading:

50%	project (5 parts)
20%	quizzes (10; 12 drop 2)
20%	final exam
10%	participation / active learning
100%	TOTAL

Questions about grading: In order to receive consideration, if you have any questions or concerns about grades you must contact our GTA via email within one week of a given grade release.

Office hours: Posted office hours (above) are times when we have committed to be available, with the primary purpose of providing assistance to students. If you can't make it to regularly posted office hours, send a message or check in before or after class, and we can schedule another time that's convenient.

Also, for quick questions, you can try to connect via email. However, for assistance via email you must ask a specific question! "What's wrong with my code?" isn't specific enough. (We will not do your debugging for you.)

The secret word is "mallard."

Class participation: You are expected to be an active participant in class. The more engaged you are, the more you will learn—and the more fun you'll have. This includes being prepared and attentive, responding when called on, participating in group discussion, and asking questions as appropriate. When it comes to asking questions, *please don't be shy!* There's no such thing as a "dumb" or "silly" question. If there's something you don't understand—*ask*! Asking questions helps you understand the material presented in the course. Asking questions is good for your classmates. It's almost certain that if you need clarification on some point, that there's at least one other student in the class with the same question. So help each other out—*ask*! Finally, when you ask a question you help me do a better job of explaining. If I explain something, and you still don't quite grasp it, chances are that I didn't do as good a job of explaining as I might have.

If you are not able to attend in-person classes please notify your instructor via email as soon as possible. While we are happy to grant reasonable accommodations due to documented illness or emergencies, you are responsible for making up any work you have missed.

Accommodations: In keeping with University policy, if you have a documented disability and are interested in utilizing ADA accommodations, you should contact Student Accessibility Services (SAS), the office of Disability Services on campus for students. SAS works with students and faculty in an interactive process to explore reasonable and appropriate accommodations, which are communicated to faculty in an accommodation letter.

Contact SAS: A170 Living/Learning Center; +1 802 656 7753; access@uvm.edu; or visit https://www.uvm.edu/access.

If you are entitled to use the Exam Proctoring Center, please book reservations at least four days in advance.

Defects / bonus points: As you might expect, we will deduct points on assignments, quizzes, or exams where you've made an error. It's only fair that instructors should be held to a similar standard. Accordingly, bonus points are awarded for reporting and correcting defects in instructor-written course materials.

- 1.0 point: Material defect. This includes any error whatsoever in code or solutions, or any error in writing that changes materially the sense of what is written. This also includes incorrect due dates for posted assessments.
- Up to 0.5 point: Minor defect. This includes typos, misspellings, or minor errors which do not affect materially the readability or sense of what is written. Determination of points for minor defects is at the instructor's sole discretion.

Due to the unfortunate behavior of some students who have favored the practice of "bonus point mining" over proper study, no student may earn more than three bonus points in this course.

Bonus points for any given defect are only awarded to the first student (across all sections) who correctly identifies the error *and provides a valid correction*. Bonus points are not available for any materials which are clearly marked as drafts. Bonus points are added to your final grade before assigning letter grades—a point or two may make a big difference. Happy hunting.

Academic integrity: Exams, quizzes, homework assignments, keys, specifications and rubrics are copyright protected works, unless clearly and explicitly indicated otherwise. Any unauthorized copying or distribution of protected works is a violation of federal law and may result in disciplinary action. Sharing of course materials without the specific, express approval of the instructor may be a violation of the University's Code of Academic Integrity and an act of academic dishonesty, which could result in disciplinary action. This includes, among other things, distributing course materials for the purpose of sharing or seeking solutions to homework or programming assignments.

Use of online services as a source of solutions is strictly prohibited. Work you submit for evaluation must be your own. No Chegg. No CourseHero. No ChatGPT or other AI-generated content. Any work not produced by you, where permitted with prior instructor approval, must be cited. Homework submissions may be screened and evaluated using plagiarism detection software (*e.g.*, MOSS).

By enrolling in this course, you acknowledge that you have read and understand the Code of Academic Integrity, and that you agree to abide by this code. Any suspected violations will be dealt with promptly. In a word: *Don't*.

See: https://www.uvm.edu/policies/student/studentcode.pdf for more information.

Student course evaluations: Students are warmly encouraged to complete an evaluation of the course at its conclusion. Evaluations are anonymous and confidential, and the information gained, including constructive criticisms, will be used to improve the course.

Statement on alcohol and other drugs: We want you to get the most you can out of this course. Therefore, you are expected to familiarize yourself and abide by the University's policies with regard to alcohol, cannabis, tobacco, and other drug use.[3] Please do everything you can to optimize your learning and to participate fully in this course.

[3] See: https://www.uvm.edu/sites/default/files/UVM-Policies/policies/drugandalco.pdf

CEMS Inclusion Statement: We wholeheartedly support the CEMS policy on diversity, equity, and inclusion:

> "Our intention is for CEMS to be a place where you will be treated with respect and kindness. We welcome individuals of all ages, backgrounds, beliefs, interests, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, ability—and other visible and nonvisible differences. All members of the College are expected to contribute to a respectful, welcoming and inclusive environment for every other member of the community. If you ever feel that you have been unfairly treated or judged by an instructor, a mentor, another student, or another member of the CEMS community, please let someone know. Your instructors and advisors in the CEMS Office of Student Services are available to discuss any concerns, or you can report an incident of bias through the Campus Bias Response Program."

Religious holidays: Students have the right to practice the religion of their choice. In order to receive extensions or excused absences, you should submit via email your documented religious holiday schedule for the semester within the first two weeks of class. Reasonable extensions will be granted where assignment deadlines conflict with religious holidays.

Student athletes: In order to receive extensions or excused absences, you should submit via email appropriate documentation as soon as possible, preferably within the first two weeks of class. Reasonable extensions will be granted where assignment deadlines conflict with team events or team travel.