# CSE 3318: Algorithms and Data Structures

Fall 2025

As the instructor for this course, I reserve the right to adjust this syllabus and schedule in any way that serves the educational needs of the students enrolled in this course.
- Alexandra Stefan



## Instructor Information

### Name

Alexandra Stefan

### Office Location

ERB 625

### Office Phone

There is no phone in my office.

### Email

astefan@uta.edu

### Faculty Profile

Alexandra Stefan

### Office Hours

Mon,Wed 2:40-3:10pm, Tue,Thu 12:40-1:40pm or by appointment

### Communication Guidelines

My preferred communication method is Microsoft Teams chat or UTA email (both are preferred over Canvas email).

I will respond to emails and voice messages within 24 hours or the following business day.

## Course Information

### Section Information

CSE 3318 sections 001 and 002

### Course Description

**Course Title:** Algorithms and Data Structures

Design and analysis of algorithms with an emphasis on data structures. Approaches to analyzing lower bounds on problems and upper bounds on algorithms. Classical algorithm design techniques including algorithms for sorting, searching; other operations on data structures such as hash tables, trees, heaps, graphs, strings; and dynamic programming and greedy approaches to optimization problems.

**Prerequisites:** *Intermediate Programming* (CSE 1320) and *Discrete Structures* (CSE 2315)

## Time and Place of Class Meetings

CSE 3318-001 Tuesday, Thursday 9:30am - 10:50am, NH 106

CSE 3318-002 Tuesday, Thursday 11am - 12:20pm, NH 106

### Time Zone

This course operates on Central Time. All times listed for class meeting times, exams, and assignment deadlines are in Central Time (CT).

## Classroom/Lecture Recording Policy

Faculty maintain the academic right to determine whether students are permitted to record classroom and online lectures. Recordings of classroom lectures, if permitted by the instructor or pursuant to an ADA accommodation, may only be used for academic purposes related to the specific course. They may not be used for commercial purposes or shared with non-course participants except in connection with a legal proceeding.

Recording of classroom and online lectures in this course is not allowed. Lecture recordings in Microsoft Teams will be available instead.

## Course Delivery Method

This course is designated on-campus, which means all lectures will be in person. To allow students to review the lectures after class and to be able to attend remotely (e.g. if sick), there will be an online Teams meeting during each lecture. The meeting will be recorded. However, the class is not online. **The delivery of the material will be geared towards the students attending in person and some lectures may fail to be recorded without prior notification.** Students should plan to attend all the lectures in person and only use the recording for review.

On days with bad weather, lectures may be held online only via Microsoft Teams. The instructor will send a Canvas announcement as early as possible and no later than 9am of that day.

# Student Learning Outcomes

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

1. Understand classic approaches to algorithm design (e.g. dynamic programming, greedy).
2. Understand specific algorithms and data structures that have wide applicability.
3. Be able to compare and choose the best algorithm that solves the problem under specific constraints (e.g. space or time limitation).
4. Apply math skills to compute the worst-case, best-case and average-case for space and time complexity of specific algorithms (e.g. worst-case time complexity of insertion sort)

5. Know what asymptotic notation means and be able to use the correct one to describe an algorithm's performance (e.g. use the correct notation for the lower-bound of space complexity).
6. Be able to solve recurrences.
7. Construct counterexamples (both the data and 'running' the algorithm on that data) that show that an algorithm does not have a certain property (e.g. to show that a specific sorting algorithm is not stable).
8. Improve programming skills - especially on pointers, data structures, recursion, and graphs.
    a. Write code with no memory errors. We will use Valgrind to check for these errors.

# Course Materials and Technology

## Textbook Information

All the information needed for assignments and tests will be provided in slides and/or presented during lectures.

*Optional textbook: Introduction to Algorithms, by Thomas H. Cormen, Charles E. Leiserson, Ronald E. Rivest, Clifford Stein,3rd edition (CLRS). The 2nd edition is also fine*.

## Technology & Equipment Requirements

**Microsoft Teams** will be used to record the lectures and to allow students to attend the lecture online if needed. Students attending online do not receive credit for in class activities.

A Unix environment will be used when grading C code, along with the Valgrind memory checker. All UTA students have access to the omega server which provides both. Other options are available (VM, Ubuntu, CodeSpaces). More information will be provided in the first 2 weeks of classes. Students are expected to use a debugger (of their choice) to debug their code. An online IDE such as OnlineGDB (https://www.onlinegdb.com/) is an alternative until students set up their computers.

Visit the OIT Services page for a list of Applications and Software available through UTA.

Visit the UTA Libraries Technology page for a list of items that can be checked out or used at the library.

# Assignments & Exams

## Weekly Quizzes

These are online quizzes intended to review the material covered that week. They will be open book, open notes and students will have multiple attempts at taking them. Occasionally a different activity may be used instead of a quiz.

One lowest score will be dropped.

## Coding Homework

These are mainly programming assignments. With a few exceptions, each homework will have a week to be completed from the time it is posted. Coding can be time consuming. Start early!

Homework scores cannot be dropped.

## Exams

There will be 3 exams (2 midterms and one final exam). They are not cumulative, but a few topics (such as time complexity) will be part of all exams.

Exam scores cannot be dropped.

## Expectations for Out-of-Class Study

Beyond the time required to attend each class meeting, students enrolled in this 3 credit-hour course should expect to spend at least an additional 12 hours per week of their own time in course-related activities, including reviewing class examples, reading required materials, completing assignments, preparing for exams, etc.

# Grading Information

| Assignments | Values (%) |
|---|---|
| Lecture Quizzes | 10% |
| Coding Homework | 30% |
| Exams (2 midterms, 1 final), all equal weight | 60% |
| | **Total: 100%** |

Students are expected to track their performance throughout the semester, which Canvas facilitates, and seek guidance from available sources, including the instructor, if their performance drops below satisfactory levels. Refer to the Student Support Services section below.

## Final Grade Calculations

| Earned pts Range | Letter Grade |
|---|---|
| 89.5 - 100.0 | A |
| 79.5 - 89.49 | B |
| 69.5 - 79.49 | C |
| 59.5 - 69.49 | D |
| 0 - 59.49 | F |

## Late Work Policy

### Late Work

Each Coding Homework can be submitted up to 24 hours late (one day late). There will be a 2 point penalty for each missed deadline in 1 hour increments. No submission is accepted after 24 hours. In Canvas, each Coding Homework will have 2 dates: a **due date** and an **available date**. The available date will be 24 hours after the due date and is set so to allow for late submissions.

Each student has 6 “remove penalty coupons”. One such coupon can be used to remove the late penalty (up to 24 hours) for one homework.

Except for the Coding Homework, all other assessments must be completed on time. There is no late submission for them and no bonus for early completion.

### Example

Assume a Coding Homework is due Wednesday at midnight (11:59pm). If submitted on:

- Tuesday or before that, it was submitted 24 hours or more early and it receives 5 bonus points if at least 25% of the homework is completed.
- Wednesday, it is on time.
- Thursday it is late and will receive 2 point penalty per missed deadline every hour.

## Make-Up Exams Policy

Make-up exams or any other additional work towards "improving one’s grade" will not be offered.

Students should notify the instructor as soon as possible (24 hours or more before the deadline) if an emergency prevents them from attending an exam or submitting a homework.

## Extra Credit Policy

Some bonus points are available from early submission of coding homework and unused “remove late penalty” coupons. Students should do their best to earn those points.

### Early submission bonus

Each Coding Homework submitted 24 hours or more early, will receive 5 bonus points.

Each “remove penalty coupon” unused by the end of the semester receives 2 bonus points.

The bonus points are added to the sum of all Coding Homework score before dividing to compute the Average homework score.

## Grade Grievance Policy

Any appeal of a grade in this course must follow the procedures and deadlines for grade-related grievances as published in the current University Catalog: Grades and Grading Policies.

# Course and University Policies

## Attendance Policy

Students should review the University Class Attendance Policies on the Class Attendance Policies page. The following attendance policy will be applied in this course.

Attending class sessions is a critical predictor and indicator of student success. The University of Texas at Arlington does not recognize a single attendance policy but encourages faculty to establish class-specific policies on attendance. As the instructor of this section, I will not take attendance, but some lectures will have in class work that will be graded. Students that are not

in class during the activity will receive 0 for the work on that day. There is no make-up for missed classwork.

## Generative AI Use in This Course

The use of Generative AI (GenAI) in course assignments and assessments must align with the guidelines established by the instructor. Unauthorized use of GenAI could result in breaches of academic integrity. Instructors are responsible for clearly delineating the permissible uses of GenAI in their courses, underscoring the importance of responsible and ethical application of these tools.

Community Standards within the Office of the Dean of Students articulate the university's stance on academic integrity and scholastic dishonesty. These standards extend to the use of GenAI. Unauthorized or unapproved use of GenAI in academic work falls within the scope of these policies and will be subject to the same disciplinary procedures.

As the instructor for this course, I have adopted the following policy on student use of GenAI.

**Prohibition of GenAI Use**

| Approach | Description |
|---|---|
| Prohibition of GenAI Use | In this course, the focus is on the development of independent critical thinking and the mastery of subject-specific content. To ensure that all submitted work accurately reflects personal understanding and original thought, the use of Generative AI (GenAI) tools in completing assignments or assessments is strictly prohibited. This policy supports our commitment to academic integrity and the direct measurement of each student's learning against the course's Student Learning Outcomes (SLOs). Any work found to be generated by AI will be subject to academic review. |

## Institutional Policies

UTA students should review the University Catalog and the Syllabus Institutional Policies page for institutional policies and contact the specific office with any questions. The institutional information includes the following policies, among others:

- Drop Policy
- Disability Accommodations
- Academic Integrity
- Electronic Communication

## UTA Honor Code

UTA students are expected to adhere to and observe standards of conduct compatible with the University's functions as an educational institution and live by the University of Texas at Arlington's Honor Code. It is the policy of The University of Texas at Arlington to uphold and support standards of personal honesty and integrity for all students consistent with the goals of a community of scholars and students seeking knowledge and responsibility.

Students are expected to complete their coding assignments on their own. All homework (coding or written) and exams are individual work by default.

If the code from two students is found to be too similar by the instructor (whether they copied from one another, or both from the same source on the internet) or if cheating is found in the exam, the case will be reported to Office of Community Standards and the following sanctions will be given:

- "-1" grade in that assignment or exam.
- Low grades due to cheating cannot be replaced.
- The student cannot receive an A in the class.

The following are considered cheating or a violation of academic honesty.

- Copying or looking at an existing solution for the entire homework or a significant component of the homework, from anywhere (a classmate, a friend, the internet, an AI generated solution). This is not allowed in any form, not even for "inspiration" or "to get an idea" or "to help me get stared".
- Any use of Generative AI in developing a homework solution
- Working together with another student to develop a single program/solution unless group or pair collaboration was explicitly allowed for that assignment.
- Copying and entire program or part of a program from another student.

- Posting your own solution to public websites or other repositories available to others.
- Giving your solution to another student or letting them look at it.

- During an exam: looking at another student's answers, exchanging information with another student, using a cheat sheet, looking at your phone, using headphones, using a smart watch or other electronic devices.

### What is allowed?

You are allowed and encouraged to discuss with classmates the homework requirements, but not specific code for the homework solution. You can practice and review programming language concepts covered in class, programs covered in class, and other practice problems that are not part of the homework. For example, if a classmate does not know how to read user input for their homework, discuss another example that uses user input, that is different from that homework.

# Student Support Services

## Student Success Center "The bugHouse"

"The bugHouse", located in ERB 570, is dedicated to supporting students.

Starting August 25th, "The bugHouse" opens its doors Monday through Friday from 10 AM to 6 PM. It is particularly useful for students in 1000 and 2000 level courses, where foundational concepts are crucial for academic success. You can seek help here with coding issues. You may be able to get help on other topics in the class, but you should be careful about that. We have very specific requirements on what methods to be used in class. If you use external sources to understand a topic, make sure it is the same method or algorithm used in class. If not, go back

to the lecture and use that knowledge to understand the material from class. Check with the instructor or a TA if in doubt.

## Student Services Page

The Student Services page provides links to many resources available to UTA students, including:

- Academic Success
- Counseling and Psychological Services (CAPS)
- Health Services
- Students with Disabilities
- Veteran Services

Students are also encouraged to check out Career Center resources to enhance their career-readiness, find student employment, search for internships, and more. We encourage Major Exploration and the use of Experiential Major Maps to keep students on track for graduation. Refer to the Graduation Help Desk for more details.

## Accessibility of Course Materials

Some course materials, such as PDFs of musical scores, technical drawings, graphs, blueprints, design plans, or artworks (common in fields like drawing, painting, or construction drafting), may not fully comply with all Web Content Accessibility Guidelines (WCAG) requirements.

The University of Texas at Arlington is dedicated to ensuring all students have equal access to information. If you experience any accessibility barriers with course materials, please know that accommodations are available. You can get assistance through the Student Access and Resource (SAR) Center or by contacting your instructor directly. Please don't hesitate to reach out if you need help.

## Online Academic Success Guide

Visit the Online Academic Success Guide to explore a list of helpful tips and resources to help you succeed in your online journey.

## UTA Health and Wellbeing Resources

UT Arlington is committed to the safety, success, and well-being of our students. To support our community, UTA has established a Community Advocacy, Response, and Engagement (CARE) Team, a dedicated group of campus professionals responsible for helping students who could benefit from academic, emotional, or psychological support, as well as those presenting risks to the health or safety of the community. If you know of someone experiencing challenges, appearing distressed, needing resources, or causing a significant disruption to the UTA community, please submit a CARE Referral by visiting the CARE Team page. You may also submit a referral for yourself if you would like additional support.

UTA students also have access to virtual, on-demand emotional support, appointment-based counseling, advanced psychiatric care, and more. For more information, visit TimelyCare.

NOTE: If a person's behavior poses an immediate threat to you or someone else, contact UTA Police at 817-272-3003 or dial 911. If you or someone you know needs to speak with a crisis counselor, please reach out to the MAVS TALK 24-hour Crisis Line at 817-272-8255 or the National Suicide and Crisis Lifeline at 988.

## Librarian to Contact

Each academic unit has access to Librarians by Academic Subject who can assist students with research projects, tutorials on plagiarism, citation references, as well as support with databases and course reserves.

# Course Schedule

| Class Date(s) | Topic(s) |
|---|---|
| Week 1 08/19, 08/21 | Introduction, Syllabus,<br>Insertion sort, linear search, binary search |
| Week 2 08/26, 08/28 | Time complexity |
| Week 3 09/02, 09/04 | Growth of functions, Summations<br>Count sort, Radix Sort, Bucket Sort |
| Week 4 09/09, 09/11 | Count sort, Radix Sort, Bucket Sort |
| Week 5 09/16, 09/18 | Mergesort, Quicksort |
| Week 6 09/23, 09/25 | Recurrences - Tree method, |
| Week 7 09/30, 10/02 | Recurrences - Master Theorem<br>**10/02 - Midterm 1 – tentative date** |
| Week 8 10/07, 10/09 | Stacks, Queues<br>Heaps |
| Week 9 10/14, 10/20 | Binary trees, BST (Binary Search Trees)<br>Leetcode problem solving |
| Week 10 10/21, 10/23 | 2-3-4 Search Tree<br>Hash Table |
| Week 11 10/28, 10/30 | **10/30 – Midterm 2 – tentative date**<br>Greedy Algorithm for Knapsack problem |
| Week 12 11/04, 11/06 | Knapsack - Dynamic Programming (DP)<br>Job Scheduling - Dynamic Programming (DP) |
| Week 13 11/11, 11/13 | DP - Longest Increasing Subsequence (LIS), Longest Common Subsequence (LCS)<br>Graphs |
| Week 14 11/18, 11/20 | Shortest Paths<br>Minimum Cost Spanning Tree |
| Week 15 11/25 | Graphs finished<br>11/27 – Thanksgiving holiday |
| Week 15 12/02<br>Finals 12/04 | Huffman Tree (Greedy Algorithm)<br>**Final Exam, 11am-12:30pm section 002, confirmed** |
| Finals 12/09 | **Final Exam, 8am-9:30pm section 001, confirmed** |

Important dates:
08/18 – First day of classes
09/01 – Labor Day Holiday
09/03 – Census date
11/26 – 11/28 – No classes. Thanksgiving Holiday
11/31 – Last day to Drop classes
12/01 – Last day of classes
12/03 – Student Study Day (no classes)