

CSE 3318: Algorithms and Data Structures

Fall 2024

Last updated 8/20/2024

Instructor Information

Instructor(s):

Alexandra Stefan (she/her)

Office Number:

ERB 625

Office Telephone Number:

817-272-3785 (CSE Department phone number)

Email Address:

astefan@uta.edu

Faculty Profile:

<http://ranger.uta.edu/~alex/>

Office Hours:

MoWe 2:45pm-3:30pm, TuTh 12:40-1:40 or by appointment. *Office hours will be in person in room ERB 625. During in-person office hours, students that are physically there will have priority. If I am free, I will check Teams and conduct online office hours (via Teams video call or chat).*

For the online office hours, I will not have an ongoing open meeting for students to join due to privacy issues. Instead, I will have individual calls with students. Please send a chat message to indicate that you want to join office hours and we can continue as a chat or with a call. For calls during office hours, I typically use video and I encourage you to use it as well, but you are free to use only voice or chat.

Course Information

Section Information, Time and Place of Class Meetings:

CSE 3318-001, TuTh 9:30am - 10:50am, face-to-face in GACB 103 and online using Microsoft Teams meetings.

CSE 3318-002, TuTh 11am - 12:20pm, face-to-face in GACB 103 and online using Microsoft Teams meetings.

The first 2 lectures may not have a recording available.

Both sections are merged in Canvas under section 001. Students from section 002 will see the Canvas course with section 001 (not their respective section) but that is simply the name used. All students from both sections, 001 and 002, will be in that Canvas course and will have normal permissions to the course.

Course Webpage:

<http://ranger.uta.edu/~alex/courses/3318/>

Description of Course Content:

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.

Student Learning Outcomes:

- Understand classic approaches to algorithm design (e.g. dynamic programming, greedy methods).
- Understand particular algorithms and data structures that have wide applicability.

- Be able to compare and choose the best algorithm that solves the problem under specific constraints (e.g. space or time limitation).
- 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)
- 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).
- Be able to solve recurrences.
- 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).
- Improve programming skills - especially on pointers, data structures, recursion, and graphs.
 - Code should not have memory errors. We will use Valgrind (already available on the Omega server and the VM) to check for these errors.

Textbooks and Other Course Materials:

There is no required textbook for this class. 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. The textbook is not required. All the needed information will be provided in slides and online lectures.

Prerequisites:

All students are expected to have passed the courses *Intermediate Programming* (CSE 1320), and *Discrete Structures* (CSE 2315).

Technology Requirements

The following online teaching tools will be used:

- Website - homework content, slides, links to additional resources
- Canvas – course announcements, online quizzes, homework submission, supplemental videos
- Teams – online lectures, office hours (for both instructor and TAs), offline discussion and questions relevant to the entire class (e.g. clarification questions for homework). We recommend using the Microsoft Teams App.
- Respondus Lockdown - software that will block your browser, may be needed.
- A webcam (integrated in the laptop or external) – may be needed.
- Headphones with microphone are encouraged, but not required.
- You can access tutorials on these tools by clicking the "Get Started" Box on your Canvas Homepage.

Other Requirements:

Students are expected to know how to use the omega server, the [VM from cse13xx](#) or Ubuntu and to write C programs and run them on these systems, to ensure same behavior as during grading. Familiarity with a [user-friendly] C-debugger is required: students will need to, and should use a good debugger when coding. Students should frequently submit and run their code on the Unix system in order to fix bugs as they appear. Do NOT submit only at the end. There may be too many bugs at that point. Options for a Unix system: the omega server, the [VM from cse13xx](#), the Ubuntu on Windows App, or a Unix/Linux machine.

Course Schedule

See the course schedule at the end of the document.

Grading Information

No Make-up Exams/Quizzes/Homework

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

Some bonus points are available from early submission of coding homework and unused extension days. Do your best to earn those.

Grading and major Assignments and examinations:

Students are expected to keep track of their performance throughout the semester (in Canvas) and seek guidance from available sources (including the instructor) if their performance drops below satisfactory levels; see “Student Support Services,” below.

See the [Grades Posting Schedule](#) for dates and deadlines related to grades.

40%	Two Midterms and a Final, in class, in person, on paper
45%	Homework (4 to 7 assignments). There will be a homework every 1-2 weeks. Do not plagiarize or collude in the homework. A grade of 0 will be given for that entire assignment and with the large homework weight it may cause you to fail the class or to receive a lower letter grade. You should not look at any solution or part of a solution for any homework. In this class, looking at a solution for “inspiration” is cheating and is penalized with grade 0 and reposted to the UTA Office of Community Standards.
15%	Weekly quizzes or activities. These will be online, open notes. In the exceptional case that a graded activity or quiz will be given during lecture time , it will be announced in Canvas at least 5 days before that lecture. 1 lowest score quiz will be dropped.
100%	Total class score (sum of above assessments)

The final grade will be based on the standard scheme: A (90-100), B (80-90), C (70-80), D (60-70). Scores are rounded to the closest integer, thus:

- 89.5 – 100 = A,
- 79.5 - 89.49 = B,
- 69.5 - 79.49 = C,
- 59.5 - 69.49 = D,
- 0 - 59.49 = F

Late and Early Submission rules for Coding Homework

Definitions:

- Available date – date until the assignment is available (i.e it can be opened in Canvas and a solution can be submitted)
- Due date – date an assignment is due. Students may not be able to submit after this date. If the Available date is after the due date, students can submit answers after the due date, but before the available date; such submissions will be late submissions and the late penalty will apply.

Bonus points for Coding Homework only

Bonus points for coding homework can be earned from:

- early submission of a homework (5 points per assignment)
 - Students that submit a Coding Homework 24 hour or more before the DUE date receive 5 bonus points for each such homework. There is no partial bonus if you submit less than 24 hours early (e.g. no bonus if you submit 20 hours before the deadline)
 - The max possible score in a coding homework will be 105 (100 points + 5 bonus points).
- unused extension days (2 pts per unused day).

If earned, these bonus points will be added to the sum of all homework scores before dividing to compute the homework average score. E.g. $HwAvg = (105hw1 + 105hw2 + 50hw3 + 10unused_days_points) / 3hws = 270 / 3 = 90$.

Bonus points from Coding Homework cannot affect other components of the grade (e.g. the Midterms and Final average score). If, with the bonus points, final homework average score, HwAvg, is above 100, it will be capped at 100.

Late submissions are allowed only for Coding Homework. Any other type of assignment has no late submission.

Coding Homework can be submitted up to 2 days (2 x 24 = 48 hours) late, after the DUE date, with 2% penalty per missed deadline every hour.

Here is an example:

- a) An assignment that can be submitted late will have both a DUE date and an AVAILABLE date in Canvas. The DUE date is the actual due date. The AVAILABLE date is 48 hours after the DUE date to allow late submission. **You cannot submit anything after the AVAILABLE date.** Coding homework submitted after DUE date, but before AVAILABLE date will be late.

- b) Late penalty: 2% of the total points for every missed deadline in one hour increments. Note that this is a PERCENTAGE, not a fixed number of points. If one homework is worth 200 points, for each missed deadline you lose 4 points, not 2 points.
- c) You have 6 “extension days”
- d) One “extension day” removes the late penalty for 24 hours.
- e) An “extension day” cannot be split. You must use an entire “extension day” or not use it at all (you cannot use 10 hours towards one assignment and another 14 towards another).
- f) No submission is allowed after 48 hours (the AVAILABLE date). You CANNOT use more than 2 extensions, or a combination of extension days with late penalty to get more than 48 hours extension from the due date. Canvas will close.
- g) The TA will apply your request for extension day(s) when they grade your homework (not before). They will manually update the number of late hours for that assignment in Canvas.
- h) **Example:**

Coding Homework X shows DUE Monday 11:59pm and AVAILABLE until Wednesday 11:59pm.

- i. Solution submitted before Sunday 11:59 pm is early. It receives 5 bonus points.
- ii. Solution submitted on Monday is on time (no bonus and no penalty).
- iii. No submission possible after Wednesday 11:59pm.
- iv. Solution submitted on Tuesday or Wednesday is late.
 1. Submitted Tuesday at 12:05 am – missed one deadline (the 11:59pm Monday) => 2% penalty
 2. Submitted Tuesday at 4:00am missed 5 deadlines (11:59pm, 12:59am, 1:59am, 2:59am, 3:59am) => 10% late penalty
 3. Submitted Tuesday at 11:50pm missed 24 deadlines => 48% penalty
 4. Submitted Wednesday at 12:05am missed 25 deadlines => 50% late penalty
 5. If you choose to use one “extension day” for this assignment, up to 24 late hours will be removed. E.g. if it was late 30 hours, now it will be late 6 hours. If it was late 18 hours, now it will be late 0 hours (on time).

6 Extension days

At the end of the semester, any unused extension day will result in 2 bonus points that will be added to the sum of all homework scores before dividing to get the average (same as early submission bonus points). E.g. 5 unused extension days result in 10 bonus points.

To keep track of the extension days, there will be an assignment in Canvas named “Extension days”. It will start by having “grade” value 6 (to represent 6 days available) and every time one or more days are used that number will be subtracted. For example, a student that used 2 extension days will have grade 4 in “Extension days” to indicate that they have 4 days left. Students are responsible for keeping track of these days. For example, if you requested to use 2 extension days for an assignment, this will not be reflected in “Extension days” for a while (it will not show until the TA graded that homework and updated the “Extension days” grade). But you should know that those days are used and count on only 4 remaining.

To use an extension day, submit a comment in Canvas saying “use 1st extension for coding hw X” under:

- “Coding Homework X” - this will tell the TA to remove the late penalty. It is mandatory.
- “Extension days” - this will help us have a centralized place with a record of what was used. This is optional. The TA will put a comment here to indicate what days were used for what homework.

I am providing the extension days to accommodate and help students handle unforeseen situations and have some flexibility in their schedule. Do not let them lure you to postpone working on your homework. I strongly encourage you to start as soon as the homework is posted and to aim to submit early. That will give you 5 bonus points for each early submission that go to the sum of all homework scores before dividing to get the average homework score. Keep the extension days and use them when you are in dire need for them (you become sick, or have some personal emergency). It is safer to keep them and not use them at all, than to use them up in the beginning and then have nothing towards the end of the semester.

Homework and Exams grading policy:

- *Problems (in homework and exam) must be solved with the specific methods covered in class unless prior permission from the instructor is granted to use a different method. The reason is that am testing your understanding of that method, not of the problem.*

- *Global variables are not allowed. Macros (created with #define) used to define specific constants are allowed (e.g. max numbers of elements in an array or max number of characters in a line of text in a file).*

Any request for re-grading (for an assignment, quiz or midterm exam) must be made within 5 business days of receipt of that grade. In case of regrading, the instructor/TA reserves the right to regrade the whole assignment or exam. After 5 business days the grade will not be modified and such requests are dismissed.

For a homework grade issue contact first the TA that graded it. They can explain why you lost points. If they made a mistake, they will fix it. If after talking to the TA the issue is not resolved, contact the instructor.

For a quiz or exam grade contact the instructor first.

IMPORTANT: Course grades will depend **EXCLUSIVELY** on the above grading criteria. Students should not request nor expect any other factor to be considered in computing the course grade. For example, the following factors will **NOT** be considered: need of a better grade to keep financial aid, to stay in the program, or to graduate. Students are expected to carefully monitor their own performance throughout the semester and seek guidance from available sources (including the instructor) if they are concerned about their performance and the course grade that they will earn. No make-up work is given at the end of the semester to “improve one’s grade”.

Coding requirements:

- Code written for a coding homework or an exam, must follow good coding style:
 - has proper and consistent indentation (3-5 spaces). Use only tabs or only spaces.
 - consistency in the placement of { }
 - good, meaningful, variable names
 - If these standards are not followed, a penalty of up to 5% will be applied or, if part of the grading criteria, the corresponding points are lost.
- Global, external, or static variables are **NOT** allowed in any code (in homework, quiz or exam) in this class. Using such variables will result in losing 50%-100% of the credit for that problem.
- Submitted homework programs must run on omega, the [VM from cse13xx](#), or the Ubuntu on Windows App. Note that your program may run on your machine, and still CRASH on omega or the VM.
- Programs that do not compile on one of the above systems receive 0 credit.
- Programs that compile, but have ANY warning receive 0 credit.
- Programs that crash may also receive 0 credit. If your program crashes at some point, only the functionality used before it crashed will be graded: no credit is given for code that does not execute.
- We will test your programs with the data provided as an example **AND WITH OTHER** test files. You are responsible for testing your programs thoroughly.

Expectations for Out-of-Class Study:

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

Before coming to class, students should have reviewed and understood the previous lecture especially in cases where the same topic is continued, or the topics are related.

Grade Grievances:

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. See [Undergraduate Grading Policies](#) and [Student Complaints](#).

Student Conduct and Academic Integrity

Students are expected to be professional and civil in their language and conduct:

- During lectures
- During office hours
- In any oral, written, or electronic communication with the instructor, TAs, and classmates
- In assignment submissions

For any student violating this policy, the instructor reserves the right to impose any grading penalties that the instructor considers appropriate, including reducing the final class grade to the next lower grade and to report the case to the Office of Community Standards. Examples of violations include language that is vulgar, insulting, disrespectful or threatening, disrupting lectures in any way, or making it difficult for other students to follow lectures in any way (e.g. by making noise or talking loud and extensive with other students at the same time the instructor or another student are speaking).

Academic Integrity:

Students enrolled all UT Arlington courses are expected to adhere to the UT Arlington Honor Code:

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

The penalty for cheating or collusion in a homework or exam is a **grade of 0 for the entire exam or homework and the case is reported to UTA's Office of Community Standards** (<https://www.uta.edu/student-affairs/community-standards>).

A student reported for academic integrity cannot receive a grade A, even if they are the one that wrote the solution and only shared it with a classmate. In this case their grade will be B, even if the class score after the cheating penalty is in the range for A. The reason for this is that a grade A represents not only academic abilities but also academic integrity.

In cases of collusion, ALL students involved are reported to the Office of Community Standards (even if one admits that he copied after the other). For example if one student wrote his/hers solution on his own, but shared it with a friend, BOTH students are reported and both are penalized with a grade of 0 for that assignment (even if one admits that he/she copied after the other student).

By default, the homework for this class is individual (no group projects) unless otherwise stated in the assignment.

You are allowed (and encouraged) to discuss with classmates the homework requirements, but NOT specifics of the homework solution. You can practice and review concepts covered in class, programs covered in class, and other practice problems that are NOT part of the homework.

You are NOT allowed to work as a team and develop together the homework solution (or a significant/critical part of it), or let another classmate see or have access to your code.

You should reference all the resources you used in preparing for a homework solution especially if they may have influenced your solution. REFRENCING MATERIAL DOES NOT JUSTIFY COPYING THAT MATERIAL. If you referenced a source, but mainly copied the code from there, that is still a violation of Academic Integrity and the same penalty is applied (grade 0). You must solve the homework and exam problems yourself, using only the materials covered in this class. You should not search and/or look at any solution (from the web, or from a friend or classmate) for homework or exam problems or part of those problems. If you need help, you should contact the instructor or a TA. You are not allowed to look at, and get inspiration from, an existing solution.

*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.*

If code or homework are stored on a cloud service, it must be private and password protected. Posting homework requirements or homework solutions on sites such as CodeHero where other users can search and access it, is a violation of the Academic Integrity policy for this class and will be reported and the grade penalty applied.

If an exam is administered in person, I may take pictures of students during the exam to document the seating arrangement. Let me know if that is a concern for you. During the exam you should only look at your exam, the board, or the ceiling. If you tend to look to the side when you think, make sure you sit at an aisle seat, where you can look towards the side wall.

Please do not hesitate to talk to me regarding any concerns you may have.

Institution Information

Please review the UTA Syllabus Institutional Policies page (<https://resources.uta.edu/provost/course-related-info/institutional-policies.php>) which covers the following policies and more. For questions, reach out to the specific office.

- Drop Policy
- Disability Accommodations
- Title IX Policy
- Academic Integrity
- Student Feedback Survey

Additional Information

Attendance:

As the instructor of this section, I may take attendance sporadically but I will not factor it into the grade. However attendance is encouraged and class participation will be factored in the course grade.

At The University of Texas at Arlington, taking attendance is not required but attendance is a critical indicator of student success. Each faculty member is free to develop his or her own methods of evaluating students' academic performance, which includes establishing course-specific policies on attendance. However, while UT Arlington does not require instructors to take attendance in their courses, the U.S. Department of Education requires that the University have a mechanism in place to mark when Federal Student Aid recipients "begin attendance in a course." UT Arlington instructors will report when students begin attendance in a course as part of the final grading process. Specifically, when assigning a student, a grade of F, faculty report must the last date a student attended their class based on evidence such as a test, participation in a class project or presentation, or an engagement online via Canvas. This date is reported to the Department of Education for federal financial aid recipients.

Emergency Exit Procedures:

Should we experience an emergency event that requires evacuation of the building, students should exit the room and move toward the nearest exit. When exiting the building during an emergency, do not take an elevator but use the stairwells instead. Faculty members and instructional staff will assist students in selecting the safest route for evacuation and will make arrangements to assist individuals with disabilities.

Student Success Programs:

UT Arlington provides a variety of resources and programs designed to help students develop academic skills, deal with personal situations, and better understand concepts and information related to their courses. Resources include [tutoring by appointment](#), [supplemental instruction](#), [academic coaching](#) (time management, study skills, etc.), [TRIO Student Support Services](#), and [student success workshops](#). For additional information, please email resources@uta.edu, or view the [Maverick Resources](#) website.

The **IDEAS Center** (<https://www.uta.edu/ideas/>) (2nd Floor of Central Library) offers **FREE tutoring and mentoring** to all students with a focus on transfer students, sophomores, veterans and others undergoing a transition to UT

Arlington. Students can drop in or check the schedule of available peer tutors at www.uta.edu/IDEAS, or call (817) 272-6593.

Supplemental Instruction (SI) leader – to be determined if there will be an SI leader for this class.

The English Writing Center (411LIBR):

The Writing Center offers **FREE** tutoring in 15-, 30-, 45-, and 60-minute face-to-face and online sessions to all UTA students on any phase of their UTA coursework. Register and make appointments online at the [Writing Center](https://uta.mywconline.com) (<https://uta.mywconline.com>). Classroom visits, workshops, and specialized services for graduate students and faculty are also available. Please see [Writing Center: OWL](#) for detailed information on all our programs and services.

The Library's 2nd floor [Academic Plaza](http://library.uta.edu/academic-plaza) (<http://library.uta.edu/academic-plaza>) offers students a central hub of support services, including IDEAS Center, University Advising Services, Transfer UTA and various college/school advising hours. Services are available during the [library's hours](#) of operation.

Librarian to Contact:

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

Emergency Phone Numbers

In case of an on-campus emergency, call the UT Arlington Police Department at **817-272-3003** (non-campus phone), **2-3003** (campus phone). You may also dial 911. Non-emergency number 817-272-3381

Library Information

Research or General Library Help

Ask for Help

- [Academic Plaza Consultation Services](http://library.uta.edu/academic-plaza) (library.uta.edu/academic-plaza)
- [Ask Us](http://ask.uta.edu/) (ask.uta.edu/)
- [Research Coaches](http://libguides.uta.edu/researchcoach) (<http://libguides.uta.edu/researchcoach>)

Resources

- [Library Tutorials](http://library.uta.edu/how-to) (library.uta.edu/how-to)
- [Subject and Course Research Guides](http://libguides.uta.edu) (libguides.uta.edu)
- [Librarians by Subject](http://library.uta.edu/subject-librarians) (library.uta.edu/subject-librarians)
- [A to Z List of Library Databases](http://libguides.uta.edu/az.php) (libguides.uta.edu/az.php)
- [Course Reserves](https://uta.summon.serialssolutions.com/#!/course_reserves) (https://uta.summon.serialssolutions.com/#!/course_reserves)
- [Study Room Reservations](http://openroom.uta.edu/) (openroom.uta.edu/)

#####

Fall 2024 Schedule - CSE 3318, 001 and 002 - Tentative

Lecture	DoW	Day	Month	Topic
1	T	20	Aug	Introduction, Syllabus, (if there is time, Examples of Algorithms-insertion sort)
2	R	22	Aug	Examples of Algorithms-binary search, insertion sort,
3	T	27	Aug	Examples of Algorithms-indirect sorting, binary search with indirect sorting
4	R	29	Aug	Time complexity
5	T	3	Sept	Time complexity
6	R	5	Sept	Growth of functions, Summations
7	T	10	Sept	Count sort, Radix Sort, Bucket Sort
8	R	12	Sept	Count sort, Radix Sort, Bucket Sort
9	T	17	Sept	Stacks, Queues
10	R	19	Sept	Heaps
11	T	24	Sept	Binary trees, BST (Binary Search Trees)
12	R	26	Sept	Leetcode problem solving
13	T	1	Oct	2-3-4 Search Tree
14	R	3	Oct	Midterm 1 - tentative
15	T	8	Oct	Hash Table
16	R	10	Oct	Optional (if time permits): Huffman Tree (Greedy Algorithms)
17	T	15	Oct	Greedy Algorithm for Knapsack problem
18	R	17	Oct	DP: Knapsack
19	T	22	Oct	DP : Job Scheduling (Greedy,DP, brute force),
20	R	24	Oct	DP : LCS, LIS, ED (other: fewest coins, rod cutting, stair climbing); DP-Memoization
21	T	29	Oct	Midterm 2 - tentative
22	R	31	Oct	Graphs
23	T	5	Nov	Minimum-Cost Spanning Trees
24	R	7	Nov	Shortest Paths, Graphs (Applications of DS in Algorithms, Language Library discussions)
25	T	12	Nov	Merge sort,
26	R	14	Nov	Quicksort
27	T	19	Nov	Recurrences - Master Theorem
28	R	21	Nov	Recurrences - Master Theorem
29	T	26	Nov	Recurrences - Master Theorem, tree method
	R	28	Nov	No class – Thanksgiving holiday
30	T	3	Dec	Last lecture
	R	5	Dec	Section 002 FINAL Exam Thursday, 11am-1pm, most likely in classroom
	T	10	Dec	Section 001 FINAL Exam Tuesday, 8am-10am, most likely in classroom

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

OTHER important dates

	4	Sept	Census date
	25	Oct	Last day to drop
	3	Dec	Last day of classes
5 to	11	Dec	Final exams week