

CSCI 4041

Algorithms and Data Structures

Spring 2019

(Last revised 12/28/2018)

Read this document carefully: it explains the structure and policies of the class, and we will assume that you know and understand all of the information contained here.

Meeting time and place:

Lecture (010): 4:00pm – 5:15pm, Monday and Wednesday, [Keller Hall 3–210](#).

Discussion:

- 011 3:35pm – 4:25pm, Thursday, [Amundson Hall 240](#).
- 012 4:40pm – 5:30pm, Thursday, [Keller Hall 2–260](#).
- 013 5:45pm – 6:35pm, Thursday, [Keller Hall 3–111](#).
- 014 6:50pm – 7:40pm, Thursday, [Ford Hall 150](#).
- 015 7:55pm – 8:45pm, Thursday, [Ford Hall 130](#).

Instructor:

Name	Nathan Taylor
Email	taylo740@umn.edu
Office	Keller Hall 6–199

For instructor and TA office hours, see the [Office Hours](#) page on the website.

Class website:

This website is the primary source of information for the Evening (010) section of 4041. The Canvas site for the class serves three purposes:

- Programming Assignments and Written Assignments will be submitted to Canvas.
- Grades will be posted to Canvas.
- Canvas has a discussion forum that students can use to form groups for the Homework and discuss course concepts.

Email Use: Most emails for the class should be sent to:

s19_4041@umn.edu

This alias will forward your message to the instructor and all of the TAs. Make sure to use Reply All when participating in an email conversation using the alias, so that all of the course staff can continue to provide input.

The exception to this is grading complaints. If you are emailing us because you think your Written Assignment was graded incorrectly, then you don't need to email the alias, just email the TA who graded you and CC me (taylo740@umn.edu). I will not interfere in a grading dispute unless the TA is clearly wrong or not following the grading guidelines I gave them for the

problem, so this is mostly between you and the TA. Below are the TA's emails:

Zechen Zhang: zhan5260@umn.edu
 Ritiz Tambi: tambi004@umn.edu
 Shi Chen: chen4595@umn.edu
 Bowen Wang: wang8330@umn.edu
 Mehrad Hajiaghabozorgi: hajia018@umn.edu

General course description: The course objective is to provide fundamental paradigms for algorithm design with the supporting data structures.

Topics covered: Rigorous analysis of algorithms/implementation. Algorithm analysis, sorting algorithms, binary trees, heaps, priority queues, heapsort, dynamic programming, greedy algorithms, graphs, graph traversal, single source shortest path, minimum cost spanning trees, binary search trees, hash tables and hashing.

[Class schedule](#)

Pre-requisites: (CSCI 1913 or CSCI 1933) and CSCI 2011

Text: Cormen, Leiserson, Rivest, and Stein: *Introduction to Algorithms* 3rd edition. ISBN-13: 978-0262033848

Additional Resources (optional):

- [1] [Algorithms, 4th Edition](#)
- [2] [Computer Science, An Interdisciplinary Approach, Chapter 4](#)
- [3] E. Horowitz and S. Sahni, "Fundamentals of Computer Algorithms", Computer Science Press, Rockville, MD, 1984.
- [4] C.H. Papadimitriou and K. Steiglitz, "Combinatorial Optimization: Algorithms and Complexity", Dover Publications, Mineola, NY, 1998.
- [5] D.E. Knuth, "The Art of Computer Programming", Addison-Wesley, Reading, MA, Volumes 1-4, Addison-Wesley, Reading, MA, 2011.
- [6] U. Manber, "Introduction to Algorithms: A Creative Approach", Addison-Wesley, Reading, MA, 1989.
- [7] A. Aho, J.E. Hopcroft, and J.D. Ullman, "Data Structures and Algorithms", Addison-Wesley, Reading, MA, 1983.
- [8] R. Lafore, "Data Structures and Algorithms in Java", Sams Publishing, Indianapolis, IN, 2002.
- [9] M. Goodrich, R. Tamassia, and M. Goldwasser, "Data Structures and Algorithms in Java", Wiley, Hoboken, NJ, 2014.
- [10] R. Sedgewick and K. Wayne, "Algorithms", Addison-Wesley, Reading, MA, 2011.

Grading: For all graded work, please address any concerns within one week of receiving the grade. After one week, no adjustments will be made. The grade breakdown for the course is as follows:

Programming Assignments (13 of them, lowest 1 dropped)	24%
Written Assignments (13 of them, lowest 1 dropped)	24%
Quizzes (7 of them, lowest 1 dropped)	30%
Recitations (14 of them, lowest 2 dropped)	12%
In-class Exercises (48 of them, lowest 8 dropped)	10%

Final grades will be assigned based the following scale:

93.0% -- 100.0%	A
90.0% -- 93.0%	A-
87.0% -- 90.0%	B+
83.0% -- 87.0%	B
80.0% -- 83.0%	B-
77.0% -- 80.0%	C+
73.0% -- 77.0%	C
70.0% -- 73.0%	C-
67.0% -- 70.0%	D+
60.0% -- 67.0%	D
0.0% -- 60.0%	F

Grades will not be rounded. For S/N grading, a satisfactory grade (S) requires a grade of 70.0% or above.

Programming Assignments:

- Programming Assignments generally consist of implementing a data structure/algorithm in some (possibly) relevant context.

- Programming Assignments will be posted each Tuesday, due the next Friday at 1:00 PM (that is, 10 days from being assigned).
- You must submit your Programming Assignment to the appropriate link on Canvas before the deadline.
- Late Programming Assignments will not be accepted, but your lowest score will be dropped.
- Programming Assignments are not group work: they must be completed individually.
- Programming Assignments must be completed in Python 3 (any 3.x version should work, but NOT Python 2.x).
- A template Python file will be provided for each Programming Assignment.
- You must implement the required data structure/algorithm as specified in the template.
- The template will include test cases for your program. You will be graded based on how many of those test cases pass.
- There will also be at least one secret test case not included in the template to prevent hard coding.
- The grading will be done automatically by a script using the output of the test cases.
- For the above reason, you must not alter anything in the template except where specified.

Written Assignments:

- Written Assignments generally consist of a set of conceptual questions or proofs, often taken from the textbook.
- Written Assignments will be posted each Tuesday, due the next Tuesday at 1:00 PM (that is, 7 days from being assigned).
- You must submit your Written Assignment to the appropriate link on Canvas before the deadline.
- Late Written Assignments will not be accepted, but your lowest score will be dropped.
- Written Assignments can be completed in groups of 1–3 students.
- Written Assignments must be submitted in PDF format, and must include the name and x500 of each group member.
- Submitting typed Written Assignments is preferred, but handwritten documents that have been scanned in are acceptable so long as the scanned PDF is still legible.
- The TAs will grade Written Assignments primarily based on the quality of your explanations, not on whether your final answer is correct.

Quizzes:

- Every other Monday, there will be a 30 minute Quiz during lecture.
- Quizzes are theoretically comprehensive, but in practice will heavily focus on the content covered since the last quiz.
- The quizzes are closed book and are not collaborative.
- You will be permitted one double-sided 8.5 x 11 sheet of handwritten notes for each quiz. Typed notes or other printed material is not acceptable.
- No make-up quizzes will be permitted, but your lowest quiz score will be dropped.

Recitations:

- Each recitation will begin with the TA discussing course content or going over examples..
- The remaining time will be allotted to group work on a set of problems related to that content.
- Groups of 3–4 students will be assigned randomly by the TA for the group work portion of recitation.
- Groups must signal the TA to check their answers to each problem as they complete them.
- Groups must get each problem checked off before moving on to the next one: splitting the group to work on different problems is not allowed.
- Students who arrive late to recitation or do not substantially contribute to their group's efforts will not receive any credit for the recitation, regardless of their group's score.
- No make-up recitations will be permitted, but your lowest two scores will be dropped.

In-class Exercises:

- Each lecture will include two in-class exercises: one at the beginning of class, and one about halfway through.
- Lectures that contain a quiz will only have one at the beginning of class.
- The exercise at the beginning of lecture will generally cover the assigned reading for the day.
- The exercise(s) during lecture will generally cover the lecture content discussed thus far.
- In-class exercises are open-book, open-notes, and fully collaborative: you are encouraged to discuss with your neighbors.
- In-class exercises will generally consist of a single multiple-choice question asked using Google Forms.
- The link for the Google Form will be posted in lecture at the start of the exercise, and will close ~5 minutes later.
- Students who are not able to access Google Forms during lecture will be given a notecard on which to submit their answer.
- In-class exercises are graded out of 1 point: your answer is either correct or it is wrong.
- No make-up in-class exercises will be allowed, but your lowest eight scores will be dropped.

Scholastic conduct:

First, [here](#) is the departmental policy on Academic Dishonesty.

Where my policy differs or further specifies the department's:

1. You are permitted to use course material without citing: this includes lecture slides, the assigned textbook, anything from your discussion sections, and any additional material I post online for THIS iteration of CSCI 4041 (you can't use material from previous semesters). This is just to avoid having to cite the textbook for every other problem.
2. I consider actively searching for a currently assigned Programming or Written Assignment problem on the internet to be Academic Dishonesty. This isn't actually enforceable, but it inevitably leads to other sorts of Academic Dishonesty when you attempt to use that material.
3. If you submit a solution that you do not understand because it came from an outside source, that is Academic Dishonesty, because it is not an accurate representation of your own knowledge of the content. This can include not understanding your group's solution if you're working in a group for the Written Assignment (make sure that everyone understands your answers before submitting), or trying to paraphrase a solution you found on the internet to avoid directly copying.
4. I'm not going to even try to enforce 6(b): Revealing or hinting to another student all or part of the essential idea(s) or solution architecture or design needed to be invented to solve any part of or all of an active grade-related assignment. Ultimately, this happens all of the time in office hours because it's just way too inefficient to insist on ensuring that only one student can hear you at a time. You're free to discuss general ideas so long as you don't actually share code on the Programming Assignment or answer write-ups on the Written Assignments.
5. More details on specific types of problems: On Programming Assignments: you can search the internet for code to help you translate a particular operation that you know how to do in Pseudocode into Python, so long as it isn't giving you a significant portion of the algorithm you're trying to implement. For example, searching how the `range()` function works, how to add an object to a list, or what a given error means is fair game: searching how to implement a priority queue in Python is not. Taking Python code for a problem that you found on the internet or from another student and changing the variable names or the loop structure does not constitute original work, and is still Academic Dishonesty. If you need help figuring out how to implement entire chunks of an algorithm, email the TA alias or come to office hours, and we'll give you guidance without directly telling you the answer, so that you hopefully learn something.

On Written Assignments, what constitutes a "significant" amount of work taken from the internet isn't clear, since it depends on the problem. Essentially, the idea is that if you are avoiding a substantial amount of work by using a solution you found online, it's Academic Dishonesty.

For straightforward problems (#1 and #4 on most assignments) where you are just told to emulate a given algorithm, you are permitted to search for other examples of the same type of problem, since they will use different values within the problem. You are not permitted to use any automated process you find online to directly solve the specific problem you face (i.e. downloading someone else's code that implements the Algorithm and just plugging in the numbers). These will most likely not be used as evidence for cheating, since there is only one correct answer, and usually only one correct way to do it.

For "find a counterexample" type problems, we shouldn't find that you regularly have the exact counterexample found in an online solution. This will occasionally happen due to sheer coincidence, but it's only used as supporting evidence and isn't grounds for a cheating accusation on its own, so don't go searching for online counterexamples just to make sure you're not matching them on accident.

For "write an algorithm" problems, don't search for the pseudocode solution or even the solution strategy on the internet. I am aware that this is somewhat inconsistent with allowing general idea discussion among students, but I am more lenient on collaboration between students (even those not in a group) than collaboration with the internet because I think you are much more likely to understand the solution if you talk it over with another student than look it up on the internet. Here there is a bit more variation than in the more straightforward problems, so these can be used as grounds for Academic Dishonesty, but there's still not a lot of variety in "correct" answers, so unless you're doing something really strange that's also found in an online solution, it probably won't be enough on its own.

For proof/explanation type problems, you are permitted to look for other examples of the same type of proof (other loop invariant proofs, time complexity proofs for different algorithms, etc.), but don't look for proofs to problems that are so similar that copy-pasting in the answer would give you most of the proof on its own. These types of problem are where we find the majority of the evidence for Academic Dishonesty, because student solutions vary wildly, so copying sticks out. How you explain something, the words that you use, the structure of the proof: none of these alone are good evidence that you used an online solution, but when all of them show similarities then it constitutes grounds for a cheating accusation.

University policies:

[Grade definitions from the Administrative Policy](#)

[Make up work for legitimate absences](#)

[Sexual Harassment, Sexual Assault, Stalking and Relationship Violence](#)

[Equity, Diversity, Equal Employment Opportunity, and Affirmative Action](#)

[Academic Freedom](#)

Incompletes: will be given only in very rare instances when an unforeseeable event causes a student who has completed all the coursework to date to be unable to complete a small portion of the work (typically the final assignment or exam). Incompletes will not be awarded for foreseeable events including a heavy course load or a poorer-than-expected performance. Verifiable documentations must be provided for the incomplete to be granted, and arrangements for the incomplete should be made as soon as such the unforeseeable event is apparent.

Disability Accommodations: We desire to make learning rewarding and fun for all students and make every attempt to accommodate anyone who has a desire to learn. If you require special classroom or test-taking accommodations, you need to contact the University Disability Services and also notify the instructor as soon as possible at the start of the semester (no later than 1 week prior to the first quiz).

Students Mental Health and Stress Managment: As a student you may experience a range of issues that can cause barriers to learning, such as strained relationships, increased anxiety, alcohol/drug problems, feeling down, difficulty concentrating and/or lack of motivation. These mental health concerns or stressful events may lead to diminished academic performance or reduce a student's ability to participate in daily activities. University of Minnesota services are available to assist you with addressing these and other concerns you may be experiencing. You can learn more about the broad range of confidential mental health services available on campus via <http://www.mentalhealth.umn.edu/>.

- © 2019 Regents of the University of Minnesota. All rights reserved.
- The University of Minnesota is an equal opportunity educator and employer
- Last modified on January 12, 2019