CSCI 310 Advanced Algorithms Syllabus (Fall 2026)

Contents

1 Logistics

1.1 Instructional Staff

Instructor: Michael Levet (He/Him/His); lastnamefirstinitial (at) cofc (dot) edu.

1.2 Key Dates

Last Day to Drop Before Grade of 'W' Is Recorded: Friday, August 21.
Last Day to Drop Before with Grade of 'W': Thursday, October 22.
Breaks: October 12-13 (Fall Break); November 25-November 29 (Thanksgiving Break). Note that the last full day of classes for the semester is Tuesday December 1.

1.3 Course Website

All announcements will be posted to the course website: `https://michaellevet.github.io/F26/CSCI310/index.html`. Students are responsible for checking the course website daily. Assignments will be posted to OAKS. Course materials will otherwise be posted to the Google Drive.

1.4 Lecture

TR 11:20-12:35, HWEA 301

I will attempt to provide a Zoom (remote synchronous) option for class as well as record lectures via Zoom, to the extent that technology cooperates. This is provided *as-is*, and students are responsible for the content covered in class regardless of whether technology cooperates. In particular, the instructor is **not** responsible if technology does not cooperate.

1.5 Office Hours

Office hours will be on Zoom. The Zoom link and days/times for office hours will be posted to my course homepage. Your success is my top priority– if any of these times don't work, please do not hesitate to email me to schedule an appointment! **If you have COVID or another contagious illness, please do not attend my office hours in-person. I will be happy to facilitate remote participation.**

2 Course Description

2.1 Prerequisites

The prerequisites include CSCI 230 Data Structures, Math 207 Discrete Structures I, and Math 120 Calculus I. This course relies **heavily** on **all** of the prerequisites. It is **dangerous** to take this course without a solid handle on the prerequisites. On the other hand, students from outside of Computer Science who are comfortable writing mathematical proofs are likely to be well prepared and are very welcome. Please discuss ASAP with the instructor if you have concerns about your background.

- CSCI 230 Data Structures (Grade of C- or Better).
- Math 207 Discrete Structures I (Grade of C- or Better).
- Math 120 Calculus I.

Note that CSCI 230 requires as a prerequisite Math 207 with a grade of C- or better. Note that Math 207 has as prerequisite Math 105, Math 111, or Math 120, which includes precalculus topics such as functions, exponentials, and logarithms. While I am very happy to answer questions outside of class, such as in office hours or over email, please be aware that course time will not be devoted to reviewing these precalculus skills.

2.2 Workload

CSCI 310 is a 3-credit course. Well-prepared students should expect to spend on average 9-12 hours/week outside of class. Students who have significant gaps in their backgrounds may find that they need to carve out additional time to review the prerequisite material. I am happy to discuss prerequisite content in office hours- please do not hesitate to reach out!

2.3 Course Content

CSCI 310 Advanced Algorithms is an undergraduate course in theoretical computer science. The primary goals include surveying fundamental algorithm design techniques, analyzing algortihm runtime complexities, and identifying computational problems that are unlikely to have efficient algorithmic solutions. We will begin the semester with a survey of including greedy algorithms, including shortest path problems, computing minimum-weight spanning trees, and network flows. Afterwards, we will discuss the technicalities of analyzing an algorithm's efficiency, including asymptotic notation (e.g., Big-O), and techniques to ascertain and compare the asymptotic runtimes (e.g., Calculus techniques, Recurrences). Once we have a sense of how to analyze algorithms, we will proceed to discuss both the divide & conquer and dynamic programming paradigms. We will also examine our algorithm design techniques closely, discussing both instances where they apply and where they fail to yield the desired results.

At the end of the semester, we will discuss Computational Complexity, which seeks to classify problems into complexity classes based on how efficiently they can be solved. The goal then is to compare these complexity classes, as opposed to individual problems. We will restrict attention to the complexity classes P (the set of decision problems that have efficient solutions) and NP (the set of decision problems where correct solutions can be verified efficiently). While it is known that $\mathsf{P} \subseteq \mathsf{NP}$, determining whether $\mathsf{P} = \mathsf{NP}$ remains the central open problem in Computer Science and one of the six biggest open problems in Math. Resolving the P vs. NP problem will have far-reaching, real-world implications, including on the security of online transactions (cryptography), curing cancer (protein folding), scheduling, routing, and a host of other combinatorial optimization problems of practical interest. Our goal will be to understand the statement of the P vs. NP problem, including contextualizing the role that our algorithmic techniques play. Our discussions on the structure of these complexity classes will be quite shallow.

We will briefly discuss Hash Tables at the end of the course.

Ultimately, this course is mathematical in nature. The obvious connections are with Discrete Math (Math 207 and 307) and Theoretical Computer Science (CSCI 410/616 Automata Theory). However, our algorithmic techniques also serve as key tools in application areas, including Artificial Intelligence (CSCI 470), Machine Learning (CSCI 334/DATA 534), Bioinformatics, Economics (ECON 324- Game Theory), Operations Research (Math 451/452), and Circuit Design (CSCI 350). In order to understand how to adapt and apply our techniques (in this course, subsequent courses, job interviews, or your careers), it is necessary to understand how and why these techniques work. For this reason, formal proofs and the underlying ideas will be examined in great detail. Therefore, a key objective in this course is to develop your mathematical maturity; that is, your ability to understand mathematical statements and formulate rigorous mathematical proofs. This will ultimately be the best indicator for success (ouside of hard work). We will rigorously prove mathematical statements in class and discuss proof strategy throughout this course. Every student will be expected to formulate proofs on homework and assessments.

Remark 1. I would recommend studying for CSCI 310 in a similar manner as Math 207/307 Discrete Structures I-II. While the material we cover has a myriad of applications, the focus will be on developing and understanding the techniques rather than on the applications themselves. The goal will be to prepare students to apply the techniques we develop beyond this course.

2.4 Learning Objectives

Algorithms is one of the key maturity courses for undergraduates in Computer Science programs (the others being Operating Systems and Programming Language Concepts). The obvious course objective is gaining proficiency with the material outlined above. Beyond that, the development of rigorous mathematical thought, mathematical maturity, and sharpness of proof writing will be emphasized. The underlying goal is for you to improve your ability to read and write mathematics, as well as appreciate the design and usage of axioms in a theoretical discipline. A third goal is to provide a solid preparation for subsequent courses that utilize rigorous algorithmic techniques. To this end, we have the following learning objectives.

- Students will work through key algorithms by hand, including Breadth-First Search, Dijkstra's Algorithm, Prim's Algorithm, Kruskal's Algorithm, the Ford-Fulkerson procedure, Quicksort and variations thereof.
- Students will implement an algorithm related to graphs.
- Students will prove theorems by induction.

- Students will prove theorems about greedy algorithms or problems amenable to greedy algorithms using exchange arguments.
- Students will construct functions to model algorithm runtimes, as well as determine closed-form asymptotic solutions for said functions.
- Students will design algorithms using the greedy, divide & conquer, and dynamic programming techniques.
- Students will ascertain when algorithm design techniques fail to apply, clearly justifying their reasoning.

2.5 Course Text

Course lecture notes will be provided, which will serve as the official course text. If you would like supplemental reading, there are several good options, including the texts by Cormen, Leiserson, Rivest, and Stein [CLRS09]; Kleinberg and Tardos [KT05]; and Dasgupta, Papadimitriou, and Vaziarni [DPV06].

Remark 2. Many of the algorithms we study have minor variations, which may impact the final answer or intermediary steps. The official version of the algorithms will be those presented in the lecture notes (and not in supplemental texts). You are responsible for using the version of the algorithm presented in the lecture notes.

Remark 3. I also highly recommend MIT's Open Courseware notes [MIT11] and Jeff Errickson's notes [Err] as supplemental resources. These are incredibly high-quality resources. In particular, Jeff Errickson has devoted considerable efforts to creating materials that are both accessible and useful for Algorithms students.

In contrast, there are a number of popular online resources that are actually harmful to use. Amongst the most popular of these is Geeks for Geeks. Many of their articles make subtle, but crucial errors (e.g., forgetting key base cases, incorrect arguments, etc.). These errors are not always apparent to students. Folks who use Geeks for Geeks and similar low-quality resources often find that their grades suffer heavily.

3 Course Structure and Grading

3.1 Grading Scheme

Assignments in this course will consist of Homework (Section 3.3), and Quizzes and Exams (Section 3.4). Homework will count for 10% of the final grade, and each assignment will be graded for completion (see Section 3.3 for details). Your two lowest HW assignments will be dropped from consideration; I recommend saving these for the end of the semester.

Quizzes and exams will count for the remaining 90% of the grade. These will be chunked into three units: Unit 1 Quiz Average, Unit 2 Quiz Average, and Unit 3 Quiz Average. For each $i \in \{1, 2, 3\}$, your Unit i Quiz Average will be calculated as follows. If you attempt $\geq 50\%$ of the in-class quizzes during Unit i, then your Unit i Quiz Average will be the maximum of the following two schemes. Otherwise, it will be Scheme 1.[1]

- **Scheme 1:** Each in-class quiz question and each question on the exam will count towards the denominator. Note that a quiz might have multiple questions. I will drop the lowest 10% of the in-class quiz questions (this does not include exam questions). The numerator will be the #Proficiency + #Outstanding. Precisely,
$$\text{Scheme 1} = \frac{\#\text{Proficiency} + \#\text{Outstanding}}{0.9 * \#\text{Quiz Questions} + \#\text{Unit } i \text{ Exam Questions}}.$$
- **Scheme 2:** Your Exam i score, calculated at:
$$\text{Scheme 2} = \frac{\#\text{Proficiency} + \#\text{Outstanding} + 0.4 * \#\text{Progress}}{\#\text{Unit } i \text{ Exam Questions}}.$$

The Final Exam will play the role of the Unit 3 exam.

The breakdown for the quiz/exam average is as follows:

[1] Exceptions to this will only be made in emergency situations and at the instructor's discretion; please contact the instructor as soon as possible to discuss your situation.

- Your Best Unit Quiz Average: 35% of your final grade.
- Your Second Best Unit Quiz Average: 30% of your final grade.
- Your Worst Unit Quiz Average: 25% of your final grade.

Note: This grading scheme incentivizes taking the quizzes seriously, throughout the entire semester. If you do well on the quizzes, then this provides some insulation for the exam. On the other hand, if you do poorly on the quizzes, but do well on the exam, then these poor quiz grades disappear entirely.

Final cutoffs will be awarded according to the following scale.

- A : $\geq 90\%$.
- A-: $\geq 88\%$.
- B+: $\geq 86\%$.
- B: $\geq 84\%$.
- B-: $\geq 80\%$.
- C+: $\geq 78\%$.
- C: $\geq 76\%$.
- C-: $\geq 70\%$.
- D+: $\geq 65\%$.
- D: $\geq 60\%$.
- D-: $\geq 50\%$.
- F: $< 50\%$.

Grades in the **A** range indicate strong preparedness for subsequent courses in Theoretical Computer Science and Math. Grades in the **B** range indicate a strong understanding of the mechanics and a moderate understanding of the theoretical underpinnings. Grades in the **C** range indicate comfort with the mechanics, such as how to execute the algorithms or solve procedural problems. Note that there are 22 such mechanical topics marked with a (*) in Section 3.7.

3.2 Grading for Individual Problems

Submissions will be graded for correctness and clearly articulated reasoning. It is not enough to arrive at the correct answer. Supporting work and reasoning must also be included and **easy to follow**. That is, both your work (including attention to intermediary details) and the clarity in which it is presented will be graded.

Each problem will receive a score of NA < Attempted (ATT) < Progress (PR) < Proficiency (P) $\leq$ Outstanding (O) as follows.

- *O* (**Near Perfect**/**Outstanding**): The solution was near-perfect and clearly explained. This is almost a **textbook** caliber solution.
- *P* (**Demonstrated Proficiency**): The solution demonstrated proficiency, with perhaps minor (but not content-related) errors. While there may be room to improve the exposition, the solution was relatively easy to follow.
- *PR* (**Demonstrated Progress, but Fell Short of Proficiency**): The solution demonstrated some measure of understanding, but had significant errors or was extremely difficult to understand.
- *ATT* (**Attempted**/**Significant Errors or Misconceptions**): The solution demonstrated glaring misunderstandings, had significant or fatal errors.

- *NA* (**No meaningful attempt**).

Note that scores of Proficiency and Outstanding are considered **full credit**, though scores of Proficiency and Outstanding count equally. We stress that solutions do not need to be perfect in order to demonstrate proficiency (this is more favorable than partial credit). Scores of NA, ATT, and PR count equally (in that they do not contribute towards proficiency), but denote feedback as to how close one was to demonstrating proficiency. A score of PR means that you likely have some handle on the content, but need to iron out some misconceptions or pay closer attention to details. A score of ATT is alarming, and you should attempt to fix any misunderstandings immediately. Please stop by office hours so I can help you!

3.3 Homework

Homework will be assigned regularly, with clearly posted deadlines. Late homework will not be accepted, unless prior arrangements are made or in emergency situations. The instructor reserves the right to make the final determination as to what constitutes an emergency. Please discuss with me as soon as possible if you have a situation that may warrant an extension. Please submit your homework via OAKS.

- Homework must be **typed** using the provided LaTeX template. Diagrams (e.g., graphs, trees) may be hand-drawn and embedded in the LaTeXdocument as an image and **oriented so that we do not have to rotate our screens to grade your work.** Please note that **handwritten solutions or those prepared without LaTeXwill not be graded.** Similarly, **if we have to rotate our screens to grade your work, then your wil will not be graded.**
- **Mathematical equations do not qualify as diagrams and may not be handwritten on homework.**
- Both your **name** and **student ID** must be included in the appropriate fields. You **must** include these on your assignment. **If I cannot identify whose assignment is being graded, then that assignment will not be graded.**
- The first question on every homework will be an honor code agreement. Failure to indicate that you have upheld the honor code will result in your assignment not being graded.
- Using generative AI (including, but not limited to ChatGPT) and regurgitating a solution it produces is an **honor code violation**. Again, anything you produce must be in your own words and reflect your understanding of the material.
- You are welcome to discuss the problems with your classmates, as well as reference outside resources. **Anything you submit must be in your own words and reflect your understanding of the material. You should be able to explain your solutions to the instructor, such as in an interview grading session.** If there are any questions about this, it is your responsibility to contact the instructor reasonably ahead of the submission deadline. **Looking up solutions or copying from other sources (including your classmates) is an honor code violation.** You must **cite** any resource (other than the course text or the instructor) that you use. This includes any classmates with whom you collaborate. Failure to cite your sources will be treated as an **honor code violation.** See Section 3.6.
- Posting to online forums for help (e.g., Chegg, Reddit, StackExchange, etc.) is an **honor code violation.** See Section 3.6.
- Individual assignments may have additional instructions beyond the syllabus. Students are responsible for adhering to those instructions.

Unless otherwise stated on a given assignment, students who make a good-faith effort on at least 75% of the problems (rounded up to the nearest integer) will receive full credit on that assignment. It is not necessary that your work is *correct*, but it is necessary that the submission reflects your good-faith efforts and understanding of the material. The point of homework is to provide an opportunity to practice with the material and receive timely feedback. There is little point in providing feedback to an AI-generated answer.

3.4 Quizzes and Exams

There will be regular quizzes, given almost exclusively in-class. Students will have 20 minutes to take the quiz (scaled for students with disability accommodations). Unless otherwise stated, quizzes are open-book and open-note, but are individual efforts. Consulting anyone who is not a member of the instructional staff about a quiz, which includes your classmates, tutors, generative AI (including, but not limited to ChatGPT) and posting online (e.g., Chegg, Reddit, Discord, StackExchange, etc.) constitutes an **honor code violation.** Similarly, all answers must be in your own words and reflect your understanding of the material. Copying from any resource is strictly prohibited. Use of electronics during a quiz is strictly prohibited. See Section 3.6.

There will also be two midterms and a final, given in-class.

3.5 Regrade Requests

Students have 7 days (including weekends) from when a grade was returned to request a regrade. **I am happy to fix mistakes in grading. Other regrade requests will not be considered.** When you submit a regrade, please clearly indicate the error made in grading. All regrade requests must be submitted using the Google form on the course homepage.

3.6 Honor Code

I expect students are familiar with policies pertaining to academic integrity, outlined in the Student Handbook. Much of what you will learn about mathematics and theoretical computer science will come from your discussions with your peers. You are welcome and encouraged to discuss the homework problems with each other and with me. It is expected that you work the problems by yourself first, so that you can contribute to the discussion. This policy will be changed, reluctantly, if I find it is being abused. **Your submissions must be written in your own words and reflect your understanding of the material.** Note that you are responsible for citing any resource (including other people) that are not members of the course staff, the course lecture notes, or the lectures. Posting to online forums for help (e.g., Chegg, Reddit, StackExchange, etc.) is an **honor code violation.** Regurgitating solutions from generative AI (including, but not limited to ChatGPT) is an honor code violation. If there are any questions regarding this policy, please ask the instructor.

Any acts of suspected academic dishonesty will be reported to the Office of the Dean of Students and addressed through the conduct process. Students found responsible for honor code violations will be subject to the following minimum penalties:

1. You will receive a -200% on the assignment.
2. You will be reported to the Office of Academic Integrity, which may choose to impose additional penalties.

Honor code violations may result in an XXF for the course, which carries the same weight as an F. The XX modifier denotes that the grade was received for academic integrity violations. **Please do not cheat.** It is not worth it.

3.7 Course Topics

In order to help you study, I have provided a list of topics that will be covered. Topics marked with a (*) are mechanical topics. Students who are regularly able to demonstrate proficiency on the topics marked with a (*) are likely to be in good shape to earn a C- or better.

Please note: Topics that are <u>not</u> marked with a (*) will likely require applying the relevant techniques in new ways, synthesis-based problem solving, or formulating a mathematical proof. In particular, problems for these topics will <u>not</u> be isomorphic variants of previous problems or examples.

1. Proof by Induction.
2. Greedy Algorithms: Graph Traversals (BFS/DFS). (*)
3. Greedy Algorithms: Dijkstra's Algorithm (Theory). (*)
4. Greedy Algorithms: Dijkstra's Algorithm (Implementation).

5. Greedy Algorithms: Examples where Greedy Algorithms Fail. (*)
6. Greedy Algorithms: Exchange Arguments.
7. Greedy Algorithms: Safe and Useless Edges. (*)
8. Greedy Algorithms: Kruskal's Algorithm. (*)
9. Greedy Algorithms: Prim's Algorithm. (*)
10. Greedy Algorithms: Network Flows: Terminology. (*)
11. Greedy Algorithms: Network Flows: Ford-Fulkerson. (*)
12. Greedy Algorithms: Network Flows: Reductions and Applications.
13. Asymptotics: L'Hopital's Rule (Polynomials, Polylogarithmic Functions) (*)
14. Asymptotics: Quasipolynomials & Series Convergence Tests (Exponentials, Factorials) (*)
15. Analyzing Code I: Independent Nested Loops (*)
16. Analyzing Code II: Dependent Nested Loops (*)
17. Analyzing Code III: Writing Down Recurrences (*)
18. Analyzing Recurrences I: Unrolling (*)
19. Analyzing Recurrences II: Tree Method (*)
20. Dynamic Programming: Identify the precise subproblems. (*)
21. Dynamic Programming: Write Down Recurrences
22. Dynamic Programming: Using Recurrence or Existing Structure to Solve (One or Two-Dimensional) Examples. (*)
23. Dynamic Programming: Design DP Algorithms.
24. Divide and Conquer: Principles and Algorithm Design (*)
25. Computational Complexity: Formulating Decision Problems (*)
26. Computational Complexity: Showing Problems belong to P. (*)
27. Computational Complexity: Showing Problems belong to NP. (*)
28. Computational Complexity: Structure and Consequences of P vs. NP.
29. Hashing and Collision (*)

4 Course Policies

4.1 Office Hours: Norms and Expectations

There will be a mix of in-person and online office hours (see Section 1.5 for the Zoom link). The purpose of office hours is to supplement lecture and the associated readings. In order to get the most out of office hours, we recommend the following.

- Watch the lectures and read through the lecture notes. In particular, work through the provided examples. These materials are there to help you!
- Spend some time working the problems first. Try to identify specific approaches you have made, as well as identify where you are stuck. If you are spending more than 30 minutes on a single problem without making much progress, then I strongly encourage you to seek help in office hours!

- If you wish to discuss specific work, please have it typed up so that you can share your screen on Zoom. It is very hard to help you if your work is on paper and you are holding it up to the camera.
- My goal is to provide hints about homework problems, as well as help students obtain momentum to keep working. In particular, I aim to help students arrive at the solutions on their own. It is completely normal to need time to digest a hint, and then come back to office hours with more questions! Learning CS Theory and Math is an iterative process- we encourage students to iterate!
- Please note that I will neither provide entire solutions in office hours nor grade work ahead of the due date.

Office Hours vs. Email: I am generally happy to discuss course logistics via email (e.g., scheduling appointments, etc.). However, email is usually not a conducive medium for tutoring. If you email me with a question about the homework (and you are certainly welcome to do so), I reserve the right to ask you to come to office hours with your question. Note that this does associate some risk with procrastination, in that you may not get your question answered until after the assignment due date (or after the quiz/exam). Similarly, if you email me late at night, I may not see your email until after the assignment is due. Please plan accordingly.

4.2 Late Work

Late work will **not** be accepted, unless prior arrangements have been made or in case of emergency situations. The final determination as to what constitutes an emergency rests with the instructor. Extensions can be requested using the Google form on the course homepage. I recognize that you all will frequently have competing deadlines, including for your other classes as well as personal obligations. There is not always time to meet all of one's deadlines. The way to handle these situations is to communicate reasonably in advance. For non-emergency situations, please request an extension at least 24 hours in advance. This includes for university-sponsored events (including, but not limited to, athletics). In general, I encourage you to ask for what you need. While I will in general try to be flexible for short-term extensions, do note that that requesting an extension does not guarantee that you will receive one.

In the event of an emergency situation which prohibits you from turning in work before the deadline, I reserve the right to choose how to handle the situation. For long-term emergencies, please talk to me.

Note that missing the deadline by a couple minutes is not a valid reason for late work to be accepted. Homework due dates and times will be clearly posted. OAKS is the final arbiter of what is considered on-time, for homework. Please plan accordingly.

4.3 Late Enrollments

Students who enroll in the course after the first day of class are subject to the same deadlines as the rest of the class.

4.4 Attendance

Attendance is not required and will only be taken during the first two weeks, for the purpose of attendance verification as required by CofC. Students who have not engaged with class by attending, completing assignments, or emailing me may be reported as having "never attended." If you are sick, please stay home– let me know if this is in the first two weeks, so that you do not get dropped. In particular, if you have COVID, please quarantine until such time as you are not contagious. I will be happy to facilitate remote participation in these instances. In the event that any member of the class (myself included) contracts COVID, I reserve the right to move the entire class online.

Note that ≥ 0 class sessions will be recorded via both voice and video recording. By attending and remaining in this class, the student consents to being recorded. Recorded class sessions are for instructional use only and may not be shared with anyone who is not enrolled in the class.

4.5 Modifications to the Syllabus

The instructor reserves the right to modify any of the policies in the syllabus at any time, particularly as dictated by the interests of learning and fairness. Students will not be graded any harsher than as outlined in Section 3.2.

4.6 Student Feedback

Student feedback regarding this course is welcome at any time. Those who wish to leave feedback anonymously are welcome to do so using the Google form on the course homepage. Students are also welcome to reach out to the instructor via email or in office hours to discuss their concerns.

5 Required Syllabus Statements

5.1 Religious Holidays

Campus policy regarding religious observances requires that faculty make every effort to deal reasonably and fairly with all students who, because of religious obligations, have conflicts with scheduled exams, assignments or required attendance. In this class, please contact the instructor within the first two weeks to discuss any conflicts with religious events. Short-term accommodations (e.g., a 24-48 hour extension on an assignment) can be requested in the normal course via the Extension Form. If you require more extensive accommodations, then you **must** discuss with the instructor within the first two weeks of class.

5.2 Students with Disabilities

The Center for Disability Services/SNAP is committed to assisting qualified students with disabilities achieve their academic goals by providing reasonable academic accommodations under appropriate circumstances. If you have a disability and anticipate the need for an accommodation in order to participate in this class, please connect with the Center for Disability Services/SNAP. They will assist you in getting the resources you may need to participate fully in this class. You can contact the Center for Disability Services/SNAP office at 843.953.1431 or at snap@cofc.edu. You can find additional information and request academic accommodations at the Center for Disability Services/SNAP website.

If you are not registered with SNAP and believe you may need a disability accommodation, please do not hesitate to contact me.

5.3 Inclement Weather, Pandemic or Substantial Interruption of Instruction

In the event of inclement weather, I will communicate a detailed plan for how class will proceed (if at all). Please prioritize your safety in these situations, including any need to evacuate. If there is a need to evacuate, I will also be prioritizing my own evacuation. The university has allocated make-up days on the weekends to be used if class is canceled for inclement weather. I will communicate in a timely manner for if/how these days will be used.

In the event of a surge in the ongoing COVID pandemic, I reserve the right to make adjustments to the structure of the class. In particular, if there exists at least one member of the class with COVID, I reserve the right to move the course online.

5.4 OAKS

OAKS will be used this semester to provide the syllabus, assignments, and such. I reserve the right to use the OAKS gradebook.

5.5 Digital Accessibility Statement

The instructor will make a good-faith effort to provide digitally accessible materials. Students who have concerns or additional needs should discuss with the instructor.

References

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to algorithms, third edition*, 3rd ed., The MIT Press, 2009.

[DPV06] Sanjoy Dasgupta, Christos H. Papadimitriou, and Umesh Vazirani, *Algorithms*, 1 ed., McGraw-Hill, Inc., USA, 2006.

[Err] Jeff Errickson, *Algorithms homepage*, Available at `https://jeffe.cs.illinois.edu/teaching/algorithms/`.

[KT05] Jon Kleinberg and Eva Tardos, *Algorithm design*, Addison-Wesley Longman Publishing Co., Inc., USA, 2005.

[MIT11] *Mit open courseware algorithms lecture notes*, 2011, Available at `https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/lecture-notes/`.