CSCI 3104 Algorithms (both sections) Syllabus (Spring 2025)

Contents

1 Logistics

1.1 Algorithms Team

- **Instructors:**

  Shamal Shaikh, shamal.shaikh at colorado.edu, section 100, TR 9:30am-10:45am in MBE 155

  Dr. Divya Vernerey, divya.vernerey at colorado.edu, section 200, MWF 11:15am-12:05pm in ECCS 201

- **Graduate Teaching Assistants (TAs):**

  Section 100: Carlina Wharton-Bucher , Rick Nueve, Peter Ly, Dhamma Kimpara

  Section 200: Mary Monroe, Elise Tate, Ali Marashian, Sourav Chakraborty

- **Undergraduate Course Assistants (CA):**

  Jerry Li, Jason Gao, Aniket Chauhan, Kaile Suoo, Isabelle Godfrey, Jonathon Lunney Please **do not** contact CAs personally about the course outside of Student Hours.

- **Graduate Graders:** Ayushiarun Sabnis, Rishiraj Paul Chowdhury, Saivenu Kolli (do not contact)

- **Graduate Course Managers:** Nirmal Khedkar and Aditi Kamatgi (do not contact)

1.2 Student Duties and Responsibilities:

- Arrive on time.
- Use laptops only for legitimate class activities (note-taking, assigned tasks).
- Do not leave class early without okaying it with the instructor in advance.
- Ask questions if you are confused.
- Try not to distract or annoy your classmates.

1.3 Canvas Course

All course material (lectures notes, video recordings), announcements, due dates, etc., will be posted to the Canvas course.

https://canvas.colorado.edu/courses/117891.

If you are enrolled in the class you should be automatically enrolled for the Canvas instance; if you are not, please contact your Instructor by email. Students are responsible for checking the Canvas announcements (either by auto-email settings or by visiting the Canvas page) regularly.

1.4 Piazza

This term we will be using Piazza for **homework/algorithm discussion and administrative clarification only**. Please only send **private messages to the Instructor** in Piazza if it concerns **grading or private in nature**.

<u>Piazza house rules:</u>

- Focused discussion on our specific topic. Please, be prepared to honor this meeting norm.
- Build on one another's comments; work toward shared understanding.
- Speak from your own experience, **without generalizing**.
- Do not offer opinions without supporting evidence.

- Take responsibility for the quality of the discussion.
- Do not monopolize discussion.
- Do not post full solutions.

2 Course Description

2.1 Prerequisites

The prerequisites include Differential and Integral Calculus, Data Structures, and Discrete Math courses. This course relies **heavily** on **all** of the prerequisites. 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.

- Differential and Integral Calculus (Grade of C- or Better in the following: MATH 1300/APPM 1350 and MATH 2300/APPM 1360)
- Data Structures (Grade of C- or Better in one of the following: CSCI 2270 or CSCI 2275)
- Discrete Math (Grade of C- or Better in one of the following: CSCI 2824, MATH 2001, APPM 3170, or ECEN 2703).

2.2 Workload

CSCI 3104 is a **4-credit course**. Following CU's guidelines, well-prepared students should **expect to spend on average 6-9 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.

Algorithms (and Theory/Math courses in general) require more time to gain traction than applied/coding-based courses. Poor early performance is not indicative of one's ability to succeed (or even earn an A) in this course.

2.3 Course Content

CSCI 3104 Algorithms is an undergraduate course in **theoretical computer science**. The primary goals include surveying fundamental algorithm design techniques, analyzing algorithm runtime complexities, and identifying computational problems that are unlikely to have efficient algorithmic solutions.

We will begin the semester with proof of correctness (that is, how do we know a given algorithm is correct?). We will then 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).

Afterwards, a survey of including greedy algorithms, including shortest path problems, and computing minimum-weight spanning trees. Then we talk about network flow problems and introduce an important network flow optimization algorithm. 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. We will then briefly talk about some advanced data structures, including Hash tables, doubling lists and its amortized analysis.

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.

Ultimately, this course is mathematical in nature. The obvious connections are with Discrete Math (MATH 3140, MATH 3170, MATH 4440) and Theoretical Computer Science (CSCI 3434, CSCI 3090, CSCI 4114). However, our algorithmic techniques also serve as key tools in application areas, including Artificial Intelligence (CSCI 3202), Machine Learning (CSCI 3832, CSCI 4622), Bioinformatics (CSCI 4314), Network Science (CSCI 3352), Economics (CSCI 7000 Algorithmic Game Theory, ECON 4050), Operations Research (CSCI 5654), and Circuit Design (ECEN 2350). 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 (outside 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. CSCI 3104 is effectively an abstract math course. This is **not** a software engineering/coding course. We also stress that 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. Our goal will be to prepare students to apply the techniques we develop beyond this course. 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.

2.4 Learning Objectives

Algorithms is one of the key maturity courses for undergraduates in Computer Science programs (the others being Systems and Principles of Programming Languages). 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. (A fun read by Josh Cooper on "Why do we have to learn proofs!?" is a must.) The underlying goal is for you to (1) improve your ability to read and write mathematics, as well as (2) appreciate the design and usage of axioms in a theoretical discipline. (3) 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 prove theorems by induction.
- 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, Mergesort, Quicksort and variations thereof.
- 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.
- Students will begin to think critically about the ethical ramifications of the design choices & applications of algorithms.

2.5 Course Text

- The classic by Cormen, Leiserson, Rivest, and Stein [CLRS09] is a good encyclopedic reference of about 1300 pages, and will serve as the **official course text**.

- We also recommend the book by Kleinberg & Tardos [KT05].
- We also recommend MIT's Open Courseware notes [MIT11].
- And you can use Jeff Errickson's notes [Eri] as supplemental resources. Errickson has devoted considerable efforts to creating materials that are both accessible and useful for Algorithms students.

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

In contrast, there are a number of popular online resources can actually be 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.

3 Course Structure and Grading

3.1 Grading Scheme

The grading breakdown is as follows:

- Homework .... 40 %
- Quiz .... 36 %
- Final Exam .... 24 %
- Extra Credit: lecture attendance ...... 1 %
- Extra Credit: recitation attendance... 1 %

Your final grade letter is determined as follows:

A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F
$\geq$ 95%	90-95%	87-90%	83-87%	80-83%	77-80%	73-77%	70-73%	67-70%	63-67%	60-63%	$<$ 60%

Roughly speaking: 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.

3.2 Key Concepts

1. Proof by induction.
2. Asymptotics: Calculus I and II Techniques (Polynomials, Polylogarithmic Functions, L'Hopital's Rule, Exponentials, Factorials, Quasipolynomials, Ratio and Root Test)
3. Analyzing Code I: Independent nested loops
4. Analyzing Code II: Dependent nested loops
5. (Greedy) Graph traversals: BFS/DFS.
6. (Greedy) Shortest path algorithms: Dijkstra's algorithm.
7. (Greedy) Examples where greedy algorithms fail.
8. (Greedy) Correctness: Exchange arguments.

9. (Greedy) Minimum-weight spanning trees: safe and useless edges.
10. (Greedy) Minimum-weight spanning trees: Kruskal's and Prim's algorithm.
11. (Greedy) Huffman coding.
12. (Greedy) Network flows: terminology.
13. (Greedy) Network Flows: Ford–Fulkerson Algorithm.
14. Analyzing Code III: Writing down recurrences
15. Analyzing Recurrences I: Unrolling
16. Analyzing Recurrences II: Tree method
17. (Divide & Conquer) Basics & counterexamples
18. (Divide & Conquer) Mergesort
19. (Divide & Conquer) Quicksort, modifications, and analysis
20. (Dynamic Programming) Identify the precise subproblems.
21. (Dynamic Programming) Write down recurrences
22. (Dynamic Programming) Using recurrence to solve.
23. (Dynamic Programming) Backtracking to find solutions.
24. (Dynamic Programming) Design dynamic programming algorithms.
25. (Data Structures) Hash tables, collisions
26. (Data Structures) Doubling lists & their amortized analysis
27. (Computational Complexity) Formulating Decision Problems
28. (Computational Complexity) Showing problems belong to P.
29. (Computational Complexity) Showing problems belong to NP.
30. (Computational Complexity) Structures and consequences of P vs NP.

3.3 Grading

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.

3.4 Extra Credit for Lecture Attendance

Attendance is required during lectures, as we will use Clickers for active learning. **Students must attend the lecture section they are registered for**, else they will not receive the Clicker points. Clicker is a tool for student engagement. It is a polling software to promote active learning by conducting live polls in class, and assess students' understanding of the material with formative assessment questions.

3.5 Extra Credit for Recitation Attendance

Attendance at recitations is required. Because recitations involve active learning activities and because of limited room sizes, we need to keep recitation sections relatively small. Towards that end: **students must attend the recitation section they are registered for.** The main theme of recitation is solving problems in groups (often using think-pair-share). Explaining the material to others serves to reinforce the material for you. Conversely, it is beneficial to have multiple perspectives on the material. It is highly likely that you will benefit from the perspectives your classmates bring to the table. Working in groups is **required**. Students who are not engaged in the material may not receive credit for that day.

Note that students **must** attend the recitations for which they are enrolled.

3.6 Homework

Homework will be assigned regularly, with clearly posted deadlines, at least a week from when the homework is released. You are responsible for being aware of both the **dates** and **times** for these deadlines. **In general, homeworks will be released on Thursdays by 11:59 pm and due the following Thursday at 11:59 pm.**

Late homework will not be accepted, except in Exceptional Circumstances. No late submissions will be accepted once solutions have been released.

Please submit your homework via Canvas.

- There will be 10 homework assignments for the semester, but we will drop the lowest two scores to calculate the final grade.
- Each homework must be submitted as a single PDF file on Canvas. The written homework must be **typed** using LaTeX. 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 it may not be graded.**
- Both your **name** and **student ID** must be included in the appropriate fields. You **must** include these on your assignment; otherwise, Gradescope may not be able to match your assignment to your submission. In this case, your submission will not be graded.
- The first question on every homework will be an honor pledge. Failure to indicate that you have upheld the honor code will result in your assignment not being graded.
- 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 (and not the TAs) reasonably ahead of the submission deadline. **Copying from other sources (including your classmates) is an honor code violation.** You must **cite** any resource (other than the course text, Instructor, TAs, or CAs) 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 5.5.
- Posting to online forums for help (e.g., ChatGPT, Chegg, Discord, Reddit, StackExchange, etc.) is an **honor code violation.** See Section 5.5.
- No Generative AI Use: You may NOT use Generative AI tools on any assignments in this course. This includes entering a prompt into Generative AI and then modifying the solution and submitting it as your own. One of the key goals of this course is to cultivate critical thinking and reasoning skills that enable students to discern credible information from misinformation, particularly in the context of AI-generated content. Mastering the concepts in the course without using Generative AI allows students to build a solid foundation, empowering them to recognize inaccuracies and biases that may arise from automated

tools. While Generative AI tools can provide quick answers and insights, their mathematical reasoning can often be flawed or misleading. Relying on these tools may hinder your ability to grasp fundamental concepts and methodologies essential for effective data analysis. You should be able to independently reproduce from scratch and explain any work that you have submitted in this class when asked.

3.7 Quizzes and Cumulative Final

There will be four in-class quizzes, but we will drop the lowest quiz grade to calculate the final grade.

The cumulative final are effectively the same as largely coordinated quizzes. The purpose is to give you time to review and reflect upon the material covered.

4 Course Policies

4.1 Office Hours: Norms and Expectations

Student hours will be held online or in-person according to staff preference (see Canvas for locations, timing, and/or Zoom links). The purpose of student hours is to supplement lecture, recitation, and the associated readings with more individualized help. In order to get the most out of student 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 we strongly encourage you to seek help in student hours!
- If you wish to discuss specific work in an online student hour, please take a picture or have it typed up so that you can more easily share your screen on Zoom. It can be hard for us to help you if your work is on paper and you are holding it up to the camera.
- Our goal is to help students improve their understanding and guide students toward solutions, as well as help students obtain momentum to keep working. In particular, we 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 student hours with more questions! Learning Math is an iterative process - we encourage students to iterate!
- Please note that the course team will not provide entire solutions in student hours, nor will they grade work ahead of the due date. If a member of the course staff states that they are unable to offer more hints without giving away a solution, please respect this. Students are welcome to bring any concerns about this to the Instructors.

Office Hours vs. Piazza: The Algo course team is happy to help with homework questions via Piazza. However, while Piazza can be effective for answering brief questions, it is not usually a conducive medium for tutoring, and we recommend longer conversations be brought to Office hours.

4.2 Regrade requests

Students have **1 week** (including weekends) from when a grade was returned to request a regrade. So if a grade is released on a Tuesday, students have through the end of the day on the following Monday to raise concerns.

The only regrade requests that will be considered are those where a mistake was made in grading. In particular, all points earned (or lost) are final, unless due to a mistake made by a grader.

There are two types of regrade requests: (1) clerical (0 score given for attempted problem, or Gradescope doesn't match grade report) and (2) conceptual (everything else). Clerical requests may be handled by any course TA or grader; conceptual requests will be handled by a course Instructor. You may submit any number of clerical regrade requests.

In order for a grade dispute to be considered, students must submit a written request clearly indicating (1) the assignment, (2) the problem(s) in question, (3) a clear explanation defending the correctness of your answer(s), (4) and an indication of where a mistake in grading was suspected. When you ask us for a conceptual regrade of one question on an assignment, Instructors reserve the right to regrade the entire assignment, and your grade on those opportunities to demonstrate proficiency may go up or down. Students are only allowed **5** conceptual regrade requests for the semester.

All regrade requests must be made through the Google Form found on Canvas. We will respond to all regrade requests within **two weeks** of receiving a request the form.

TAs and CAs are not empowered to change grades without Instructor approval; please do not try to argue your grades with them.

4.3 Solutions

Solutions for Homework will be posted on Canvas, generally within a couple days of the due dates. Solutions will typically **not** be posted for the recitation activities.

4.4 Late Work

The course staff will evaluate more than 10,000 pages of homework this semester. For this to run smoothly, we need your help.

No credit will be given for assignments submitted in any other way (e.g., email) or after the deadline.

Note that missing the homework deadlines by a few minutes is not a valid reason for late work to be accepted. Homework due dates and times will be clearly posted. Please plan accordingly.

4.5 Exceptional Circumstances

In the case of exceptional circumstances, e. g., significant illness or injury that preclude submitting an assignment on time, please email your Instructor. Examples of ***unexceptional*** circumstances include registering late, travel for job interviews or conferences or fun, forgetting the homework deadline, or simply not finishing on time. Our goal is for you to become proficient in the material and to demonstrate that to us, and in exceptional circumstances we will work with you to come up with a suitable arrangement.

5 Required Syllabus Statements

5.1 Classroom Behavior

Students and faculty are responsible for maintaining an appropriate learning environment in all instructional settings, whether in person, remote, or online. Failure to adhere to such behavioral standards may be subject to discipline. Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, political affiliation, or political philosophy. For more information, see the policies on classroom behavior and the Student Conduct & Conflict Resolution policies.

5.2 Requirements for Infectious Disease

Members of the CU Boulder community and visitors to campus must follow university, department, and building health and safety requirements and all applicable campus policies and public health guidelines to reduce the risk of spreading infectious diseases. If public health conditions require, the university may also invoke related requirements for student conduct and disability accommodation that will apply to this class.

If you feel ill and think you might have COVID-19 or if you have tested positive for COVID-19, please stay home and follow the guidance of the Centers for Disease Control and Prevention (CDC) for isolation and testing. If you have been in close contact with someone who has COVID-19 but do not have any symptoms and have not tested positive for COVID-19, you do not need to stay home but should follow the guidance of the CDC for masking and testing.

5.3 Accommodation for Disabilities, Temporary Medical Conditions, and Medical Isolation

If you qualify for accommodations because of a disability, please submit your accommodation letter from Disability Services to your faculty member in a timely manner so that your needs can be addressed. Disability Services determines accommodations based on documented disabilities in the academic environment. Information on requesting accommodations is located on the Disability Services website. Contact Disability Services at 303-492-8671 or dsinfo@colorado.edu for further assistance. If you have a temporary medical condition, see Temporary Medical Conditions on the Disability Services website.

5.4 Preferred Student Names and Pronouns

CU Boulder recognizes that students' legal information doesn't always align with how they identify. Students may update their preferred names and pronouns via the student portal; those preferred names and pronouns are listed on Instructors' class rosters. In the absence of such updates, the name that appears on the class roster is the student's legal name.

5.5 Honor Code

All students enrolled in a University of Colorado Boulder course are responsible for knowing and adhering to the Honor Code academic integrity policy. Violations of the Honor Code may include, but are not limited to: plagiarism, cheating, fabrication, lying, bribery, threat, unauthorized access to academic materials, clicker fraud, submitting the same or similar work in more than one course without permission from all course instructors involved, and aiding academic dishonesty. All incidents of academic misconduct will be reported to the Honor Code (honor@colorado.edu); 303-492-5550). Students found responsible for violating the academic integrity policy will be subject to nonacademic sanctions from the Honor Code as well as academic sanctions from the faculty member. Additional information regarding the Honor Code academic integrity policy can be found on the Honor Code website.

Intellectual dishonesty or plagiarism of any form, at any level, will not be tolerated.

Discussing problems with other students is encouraged, but you must list your collaboration on the page where you give the solution. If you discussed it with 20 other people, then all 20 names should appear in your solution. If someone was particularly helpful, say so. Be generous; if you?re not sure whether someone should be included in your list of collaborators, include them. For discussions in class, in section, or in office hours, where collecting names is impractical, it?s okay to write something like ?discussions in class.? There is no penalty for discussing problems with other students.

Copying from any source in any way is strictly forbidden. This includes both the Web and **other students (past or present)**. If you are unsure about whether something is permitted, please ask the Instructors before the assignment is due.

Posting to online forums for help (e.g., ChatGTP, Chegg, Discord, Reddit, StackExchange, etc.) is an **honor code violation.**

Write everything in your own words and cite all outside resources. You are strongly encouraged to use outside resources, but you must write your solutions yourself. We are not interested in seeing Wikipedia's or anyone else's solution. The only sources you are not required to cite are the textbook and lecture notes, and the prerequisite material.

There will be a zero-tolerance policy to violations of this policy.

Any honor code violation will result in the following minimum penalties:

(a) First offense: zero for the full assignment in question (does not count as lowest drops).

(b) Second offense: zero for the full assignment in question (excluding lowest drops) and referral to the CU Honor Council.

(c) Third offense: F in the course.

Please do not cheat. It is not worth it.

5.6 Sexual Misconduct, Discrimination, Harassment and/or Related Retaliation

The University of Colorado Boulder (CU Boulder) is committed to fostering an inclusive and welcoming learning, working, and living environment. CU Boulder will not tolerate acts of sexual misconduct (harassment, exploitation, and assault), intimate partner violence (dating or domestic violence), stalking, or protected-class discrimination or harassment by or against members of our community. Individuals who believe they have been subject to misconduct or retaliatory actions for reporting a concern should contact the Office of Institutional Equity and Compliance (OIEC) at 303-492-2127 or email cureport@colorado.edu. Information about OIEC, university policies, reporting options, and the campus resources can be found on the OIEC website.

Please know that faculty and graduate instructors must inform OIEC when they are made aware of incidents related to these policies regardless of when or where something occurred. This is to ensure that individuals impacted receive outreach from OIEC about resolution options and support resources. To learn more about reporting and support for a variety of concerns, visit the Don't Ignore It page.

5.7 Religious Accommodations

Campus policy requires faculty to provide reasonable accommodations for students who, because of religious obligations, have conflicts with scheduled exams, assignments or required attendance. Please communicate the need for a religious accommodation in a timely manner. In this class, we will make reasonable efforts to accommodate such needs if you notify the professors of their specific nature by the end of the 3rd week of class (Friday, January 31, 2025).

See the campus policy regarding religious observances for full details.

5.8 Mental Health and Wellness

The University of Colorado Boulder is committed to the well-being of all students. If you are struggling with personal stressors, mental health or substance use concerns that are impacting academic or daily life, please contact Counseling and Psychiatric Services (CAPS) located in C4C or call (303) 492-2277, 24/7.

Free and unlimited telehealth is also available through Academic Live Care. The Academic Live Care site also provides information about additional wellness services on campus that are available to students.

6 Tentative Schedule

Week	**Item Due**	**Topic**
1	HW 0	Syllabus, Proof by Induction
2	HW 1	Asymptotics: Definitions, Transitivity, Limit Comparison Test
3	HW 2	Analyzing Code: Independent and Dependent Nested Loops
4	QUIZ 1 (M, T)	Breadth-First and Depth-First Search, Begin Dijkstra's Algorithm
5	HW 3	Finish Dijkstra's Algorithm, Proof of Correctness, Where Greedy Algorithm Fails
6	HW 4	Exchange Arguments; Spanning Trees, Safe and Useless Edges
7	QUIZ 2 (T, W)	Kruskal's Algorithm, Prim's Algorithm
8	HW 5	Network Flows: Terminology; begin Ford-Fulkerson
9	HW 6	Network Flows: Ford-Fulkerson; Max-Flow, Min-Cut Theorem
10	QUIZ 3 (M, T)	Analyzing Recurrence: Geometric Series, Unrolling and Tree Methods
–	Spring Break	–
11	HW 7	Divide & Conquer: Mergesort, Quicksort
12	HW 8	Dynamic Programming: Writing Down Recurrences and Identifying Subproblems
13	QUIZ 4 (T, W)	Dynamic Programming: Rod Cutting, Longest-Common Subsequence
14	HW 9	Hashing and Collisions
15	HW 10	NP-completeness; P vs. NP
–	FINALS	–

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.

[Eri] Jeff Erickson, *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/`.