CSCI 3104: Algorithms

Charles Carlson and Ewan Davies

Fall 2020

1 Essential information

	Section 100	Section 200
Instructor	Ewan Davies	Charles (Charlie) Carlson
Lectures	TTh 9:35–10:50am	TTh 2:20–3:35pm
Email	ewan.davies@colorado.edu	charles.carlson@colorado.edu
TAs	Camden Elliott-Williams, Michael Levet, Robert Green, Vishnu Murali, Abteen Ebrahimi, Yichen Wang, Wanshan Yang, Abhilash Raghavendra Jahagirdar	
CAs	Alex Book, Angel Dong, Cedric Kram, Luke Ingalls, Ian Jorquera, Erik Rhodes, Kathleen Tran	
Canvas	https://canvas.colorado.edu/courses/65183	
Course website	https://www.ewandavies.org/courses/csci3104	
Piazza	https://piazza.com/colorado/fall2020/csci3104	

Prerequisites

Data Structures or Programming and Data Structures (CSCI 2270 or CSCI 2275) and Calculus 1 & 2 (up to APPM 1360 or MATH 2300) and one of Discrete Structures (CSCI 2824), Discrete Mathematics for Computer Engineers (ECEN 2703), Discrete Applied Mathematics (APPM 3170), Introduction to Discrete Mathematics (MATH 2001). Minimum grades C−.

We cover proof by induction, recurrence relations, limits, and graphs from Math and Discrete Math, so familiarity with these topics is highly desirable.

Programming experience is desirable, especially in an imperative language such as C, C++, Python, or Java. Programming assignments are required to be submitted in Python, but extensive knowledge of Python will not be required and we believe students familiar with any imperative language will be able to adapt to Python without too much difficulty.

Recommended reading (not required)

- Introduction to Algorithms, Cormen, Leiserson, Rivest, and Stein, MIT Press 978-0-262-03384-8. Available to students for free through the Online CU Library here, and any edition suffices.
- Algorithms, Erickson, self-published 978-1-79264-483-2. Available for free online at Erickson's website `http://algorithms.wtf`.

Course objectives

- Become familiar with standard algorithms and their merits and drawbacks in practice.
- Learn to prove an algorithm's correctness and rigorously analyze time and space complexity.
- Practice adapting and combining algorithms to solve real problems.
- Learn strategies for designing new algorithms for emerging applications.
- Learn to communicate clearly about algorithm design, correctness, and complexity analysis.

Course structure

This course is delivered *synchronously, fully remote.* All lectures, recitations, and office hours will take place live on Zoom with the meeting links posted to Canvas. You will need to use zoom with your CU account to access the meetings, and possibly give a passcode to enter.

All resources and information will be posted to Canvas and the course website, and we encourage discussions on Piazza. Assignments will be submitted online through Canvas or Gradescope, and quizzes and exams will be administered online.

Lectures will be recorded and made available to view online afterwards. By attending any lecture you consent to be recorded and have the data stored on a private server accessible with your CU Boulder credentials. Do not share recorded lectures (or links to them) with people not enrolled in the course, this is a violation of FERPA Protections. We do not require recitations or office hours to be recorded, the relevant TAs and CAs will make it clear if they plan to record any part of a session.

We demand reasonable and appropriate conduct online including on Zoom and Piazza, and we will not tolerate disruption, foul language, or insults of staff and students. You may be banned from Piazza or the entire course as a result of violating this policy.

Changes

We reserve the right to alter this syllabus including the course schedule, standards, and assessment procedures to make things easier for you and us. We will endeavor to be clear and fair with any changes, but it is important that we can adapt to unfolding events.

2 Required syllabus statements

Classroom behavior

Both students and faculty are responsible for maintaining an appropriate learning environment in all instructional settings, whether in person, remote or online. Those who fail 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 Code of Conduct.

Requirements for COVID-19

As a matter of public health and safety due to the pandemic, all members of the CU Boulder community and all visitors to campus must follow university, department and building requirements, and public health orders in place to reduce the risk of spreading infectious disease. Required safety measures at CU Boulder relevant to the classroom setting include:

- maintain 6-foot distancing when possible,
- wear a face covering in public indoor spaces and outdoors while on campus consistent with state and county health orders,
- clean local work area,
- practice hand hygiene,
- follow public health orders, and
- if sick and you live off campus, do not come onto campus (unless instructed by a CU Healthcare professional), or if you live on-campus, please alert CU Boulder Medical Services.

Students who fail to adhere to these requirements will be asked to leave class, and students who do not leave class when asked or who refuse to comply with these requirements will be referred to Student Conduct and Conflict Resolution. For more information, see the policies on COVID-19 Health and Safety and classroom behavior and the Student Code of Conduct. If you require accommodation because a disability prevents you from fulfilling these safety measures, please see the "Accommodation for disabilities" statement on this syllabus.

Before returning to campus, all students must complete the COVID-19 Student Health and Expectations Course. Before coming on to campus each day, all students are required to complete a Daily Health Form. Students who have tested positive for COVID-19, have symptoms of COVID-19, or have had close contact with someone who has tested positive for or had symptoms of COVID-19 must stay home and complete the Health Questionnaire and Illness Reporting Form remotely.

In this class, if you are sick or quarantined you can request deadline extension for homework and online quizzes. As all teaching in this course is fully remote you are welcome to attend as much as you can though your quarantine period. Lectures are recorded so you can catch up when you are well again. A “doctor’s note” is not required to request a deadline extension, and you are not required to disclose confidential health information to course staff. If you have been exposed to a confirmed or presumed positive COVID-19 case you should prioritize self-isolation over administrative tasks such as obtaining a doctor’s note, and seek medial advice only if you experience symptoms. This reduces the load on our healthcare resources and allows treatment of the sick to be prioritized.

Accommodation for disabilities

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.

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. Please be patient while the course staff learn your names and graciously correct the staff if they initially use the wrong name or pronoun. If this remains a problem then please notify an instructor who will help you be addressed correctly.

The instructors of this course are happy to be addressed by their first names, without formal titles. Section 100 is taught by Ewan (he/him) and section 200 by Charlie (he/him).

Honor code

All students enrolled in a University of Colorado Boulder course are responsible for knowing and adhering to the Honor Code. Violations of the policy may include: 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 at the Honor Code Office website.

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 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 `cureport@colorado.edu`. Information about the OIEC, university policies, anonymous reporting, and the campus resources can be found on the OIEC website.

Please know that faculty and instructors have a responsibility to inform OIEC when made aware of incidents of sexual misconduct, dating and domestic violence, stalking, discrimination, harassment and/or related retaliation, to ensure that individuals impacted receive information about options for reporting and support resources.

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 there is substantial scope for such accommodations as we deliver the course online and the recordings for every lecture can be watched afterwards at your convenience. Please raise any concerns with your instructor or TA for recitations in advance of any religious holidays. See the campus policy regarding religious observances for full details.

3 Detailed information

Assessment structure

This course uses mastery-based grading, sometimes known as standards-based grading. For a brief introduction see e.g. the first 6 minutes of `https://www.youtube.com/watch?v=G587eEZjRJA`.

- There are 24 content standards (listed below) and 1 participation standard.
- For each content standard you will have several opportunities (usually at least 4) to demonstrate that you have *mastered* that standard. To be awarded a standard, you must demonstrate mastery twice.
- You earn points towards your participation standard by doing the online quizzes, by attending recitation, and by posting high quality content to Piazza. If you earn enough points, you are awarded the participation standard.
- Your final grade is determined by how many standards you are awarded (i.e. demonstrate mastery of at least twice), as follows:

No. standards awarded	≥ 21	20	19	18	17	16	15	14	13	12	11	≤ 10
Final grade	A	A−	B+	B	B−	C+	C	C−	D+	D	D−	F

You will have opportunities to demonstrate mastery each week on homework and an online quiz, on midterm exams in weeks 8 and 13, and on a final exam. This means that it is possible to attain mastery for most standards either (i) in homework and a quiz (without passing any exam), (ii) entirely in exams (a midterm plus the final), or (iii) a mixture of these modalities. We believe this offers you (our students) the best chance of achieving a high grade, as you can focus on the types of assessment that suit you.

We strongly encourage you to keep up with homework and quizzes as it is very hard to cram for this course during exam weeks. We are happy if you are awarded a standard before any exam situation, allowing you to focus your attention on any remaining standards in the exam. Note that failing to attain mastery at a given opportunity does not necessarily harm your grade at all, if you can still attain mastery twice you can still be awarded the standard. This is different than the traditional assessment structure where a C- on some assignment pulls down your average forever, and we think the mastery system leads to less stress and better learning for this reason.

Each question on an assignment or exam will be graded from 0 to 4 as follows:

0. No meaningful attempt made
1. Attempt made but demonstrates very little understanding
2. Near mastery but missing an important detail or too many small details
3. Mastery, though we allow tiny omissions or calculation errors
4. Perfection, we could use your solution as a model of how to succeed in this course

The numerical values are only for feedback and have no effect on your grade. **The only relevant thing from a grade point of view is whether you got at least a 3 (mastery), or at most a 2 (not mastery).** We strongly discourage you from regrade requests (discussed in detail below) if it is clear that your solution would not achieve at least a 3, there is no point getting a 1 upgraded to a 2 or a 3 upgraded to a 4.

Weekly assignments. Each week there will be a homework assignment and a timed online quiz. These will both contain opportunities to demonstrate mastery of some of the standards. On the homework assignment we will not say specifically which problems go with which standards (though you can probably guess pretty accurately). It is possible that a single problem could count towards more than one standard. On quizzes (and exams), each problem will be clearly marked with the standard it corresponds to, allowing you to prioritize as you see fit.

Regrade requests. On Gradescope, there is a “regrade” button, but this should only be used if you received a zero for “no meaningful attempt was made” on a given question (or part of a question), and yet you did make a meaningful attempt. Regrade requests through Gradescope for any other purpose will receive a pointer to this policy and then be marked resolved. We use the Gradescope facility simply to catch this rather obvious “missed submission” problem.

On the other hand, if you want the logic of your answer regraded then we have a specific regrade procedure. You have precisely one week from the time your score was made available to request a regrade via the google form which can be found via the link on Canvas. To fill out this form you need to provide a clear explanation of what your solution was, what rating it received (which must be at most a 2), and why you think it deserved a higher rating (which must be at least a 3). If your regrade request is granted, we reserve the right to regrade your entire problem set, and the rating on each problem might go up or go down. We discourage regrade requests in general as the mastery system offers you many chances to receive full credit for a standard.

Exams. The midterm in week 8 covers material presented during weeks 1–6, and the midterm in week 13 covers material presented during weeks 7–12. The final exam in finals week covers all material. On the exams questions will be clearly marked with which standard they correspond to. If you have already demonstrated mastery twice in some standard you do not need to attempt questions relating to that standard again.

Overall, you thus have 4 default opportunities to demonstrate mastery in each standard:

1. On a problem set
2. On an quiz
3. On a midterm
4. On the final.

There can be difficulty with the topics that we teach right at the end of the semester as there may not be enough time to allow for 4 opportunities to demonstrate mastery. We will make sure that this is not a problem by offering additional assessments or permitting some standards to be awarded after only 1 demonstration of mastery as appropriate. Note that you can still get an A without mastering every standard. Recall that we reserve the right to adjust the standards if it suits us and you, for example some may be removed and the grade boundaries adjusted appropriately if we spend too much time on other topics.

The participation standard is unique. You gain points towards this standard by attending lectures and recitations, and **bonus points are available for students who start or contribute to high-quality discussion on Piazza**. In order to foster high-quality discussion on Piazza we suggest that motivated students prepare one or more short reports that compare and contrast aspects of the algorithms that we cover in the course. For example, you could compare quicksort and mergesort, discuss how Dijkstra’s algorithm and Bellman–Ford are useful in different real-world situations, discuss how greedy graph coloring with two colors detects bipartite graphs, or many more topics. We will provide suggestions throughout the semester. Bonus participation points will be offered for any reasonably written report and for any discussion on other students' reports. We will not be carefully assessing these reports, they are intended to replace the discussions often had by groups of students that may be more difficult to have when the course is administered fully remotely. We stress that **posting discussion reports on Piazza is not mandatory** and anybody who attends a high percentage of the lectures and recitations will be awarded the participation standard.

Standards

1. Loop invariants and proof by induction
2. Asymptotic notation
3. Asymptotic analysis I: correctly writing down equations for counting operations given pseudocode with simple independent loops
4. Asymptotic analysis II: correctly writing down equations for counting operations given pseudocode with nested dependent loops
5. Recurrence relations I: write down correct recurrence for a recursive algorithm
6. Recurrence relations II: solve by unrolling
7. Recurrence relations III: solve by tree method
8. Divide and conquer: important principles (when does it apply and when not)
9. Average case analysis (of quicksort), compare to worst-case behavior
10. Hash tables: collisions, load factor, when to apply vs balanced binary tree
11. Greedy algorithms I: exchange argument for correctness (mastery can be demonstrated either using interval scheduling or Huffman coding)
12. Greedy algorithms II: examples where greedy algorithms do not work
13. Shortest paths I: breadth-first search for unweighted graphs
14. Shortest paths II: Dijkstra's algorithm (or Bellman–Ford which we cover later)
15. Minimum spanning trees I: safe and useless edges
16. Minimum spanning trees II: the generic algorithm and at least one of Kruskal's or Prim's algorithm
17. Max flow min cut I: residual graph
18. Max flow min cut II: applying max flow (reducing to max flow, e.g. bipartite matching)
19. Dynamic programming I: principles (when does it apply and when not)
20. Dynamic programming II: rigorous understanding of a simple 1-dimensional example (Fibonacci numbers or rod cutting)
21. Dynamic programming III: write down recurrence for computing local solution in a complex example (e.g. edit distance)
22. Dynamic Programming IV: order of subproblems and topological sort
23. Dynamic Programming V: backtracking (recovering the optimal solution not just its value)
24. Complexity theory: basic definitions (P, NP, NP-hard, NP-complete, reduction)
25. Participation standard

Tentative schedule

Week 1. Proof by induction, loop invariants, *insertion sort*
Week 2. Asymptotic analysis, start divide and conquer algorithms
Week 3. Divide and conquer algorithms, recurrence relations, *mergesort*, *quicksort*
Week 4. Average case analysis, *randomized quicksort*, hash tables
Week 5. Greedy algorithms I: introduction, *Huffman codes*
Week 6. Greedy algorithms II: graph search including *breadth-first search* and *Dijkstra's algorithm*, start Greedy algorithms III: minimum spanning trees including *Prim's* and *Kruskal's algorithm*
Week 7. Greedy algorithms III and IV: finish minimum spanning trees, *max flow min cut*
Week 8. Review and midterm
Week 9. Dynamic programming I: introduction, *Fibonacci numbers*, *rod cutting*
Week 10. Dynamic programming II: the *Bellman–Ford algorithm*
Week 11. Dynamic programming III: edit distance
Week 12. Complexity theory: definitions of P, NP, etc., reductions
Week 13. Review and midterm
Week 14. Complexity theory: further discussion, approximation algorithms
Week 15. Final review and non-examinable bonus material

Getting help

- Attend the lectures, attend your recitation section, and come to office hours. These are provided specifically to help you learn the material. Take advantage of them.
- Your TA will refuse to solve homework problems in office hours as this constitutes too much assistance on assessed work. You are encouraged to ask for help understanding the lectures and relevant material, and to ask for examples of the knowledge you need for homework solutions.
- Use the Piazza forum for class discussion, general questions, and even specific questions about the course content. Rather than emailing questions to the teaching staff, please post your questions on our Piazza forum. The entire class benefits if good quality discussion of our course is posted to Piazza for everyone to read. Everyone who meets the given number of standards will get the grade they deserve, appropriate discussions with your classmates cannot bring down your grade!
- Do not abuse email. Do not ask for help with homework or quizzes via email. If every student in class sent us 1 email a day, and it took only 5 minutes to respond to each (which is a hilarious underestimate of how long responding takes), the course staff would be spending over 33 hours a day just responding to email. Email should be reserved for high- priority issues only, and any private matters (illness, immigration issues, stress, etc.) that are not suitable for Piazza.
- The provided lecture notes are intended to be complete, correct, comprehensive, and mostly self-contained. Supplementary reading will greatly enhance your learning but

it is not necessary to purchase and read textbooks to pass the course. If you find an error in our lecture notes it can be a good idea to raise the issue on Piazza so that we can release corrected notes as soon as possible.

Homework

The course staff will evaluate more than 10,000 pages of homework this semester. For this to run smoothly we set strict rules about deadlines. Since the mastery-based grading allows you many chances to succeed with full credit we will not bend these rules without *very* good reasons.

- Homework solutions will be due about a week after being posted. This should usually be Thursday night and the details will be clearly posted for each assignment. Submission will either be to Canvas or Gradescope depending on the assignment. Regardless, submissions will be a single pdf file and possibly an extra plain text file for source code.
- No credit will be given for assignment submitted in any other way (e.g. email), or submitted after the deadline
- All homework submissions must be in pdf format produced by latex and conform to the template that we will release with each set of questions. We rely on Gradescope to help us with the logistics of grading, and answers that are not in the specified regions may not be considered.
- Specific help with how to produce documents in latex is available in many places online, and you can ask for help getting this done efficiently in office hours. Latex is free to download and install, and can even be used for free in a web browser with no install required (see e.g. `https://www.overleaf.com`).
- The answer templates have a space for you name and ID. If your name and ID are not in this box you will not get credit.
- Solutions must be detailed and clear. The clearer your explanation the more likely you are to receive full credit for a correct answer. Many mathematical statements are best discussed with good, precise use of English written in full sentences; we do not encourage a stream of symbols as a clear way to communicate proofs and logical arguments.

Programming assignments (of which there are only a few)

- We require you use **Python 3**. Specifying version 3 helps us all avoid silly `print` function/statement issues. We find that Python lets you focus more on the algorithm and less on details like pointers, classes, and incantations of `public static void main(string[] args)` etc.
- Your code must be runnable and readable including good variable names, formatting, and comments where appropriate.

- You must submit source files (i.e. the Python code) in addition to your solution pdf, and it must be runnable. Our assignments will not be so complex that using multiple files seems useful, we encourage single-file submissions.
- By default the Python interpreter executes line-by-line any code in the file, so you can use this to your benefit. We do not require you to develop clever, interactive command-line utilities.
- Unless specifically allowed, all parts of all algorithms and data structures must be implemented from scratch. Python has a large standard library and although we will always allow you to use `list` (which is actually an array) and `tuple`, you can only use other data structures if specified in the assignment. If you are unsure what we are trying to assess, ask for clarification on, say, Piazza as it's likely other students will want to know too.

Rules for formatting and submitting your work

1. On-time rule: late submission are not accepted.
2. Canvas/Gradescope rule: all homework and quiz submissions must be made through the class Canvas page or Gradescope as will be instructed on each assignment.
3. One-file rule: your solutions to the problem set must be a single pdf file that matches the template. The only exception to this is source code files, which should be submitted as separate text files.
4. Format rule: your submission must be a pdf and the rules are different for problem sets and for quizzes.

    - For problem sets **you must** download the template as a .tex file, rename it according to the filename rule (see below), type in your solutions in the correct places, compile (a few times) and submit the resulting pdf produced by latex.
    - For quizzes, which are timed, you have two additional options. You can download the template as a .tex file as above and include images of your hand-written solutions with `\includegraphics` in the correct parts of the template, or you can take the .pdf version of the template and use a tablet and stylus to annotate the pdf, save it, and submit the annotated pdf. Both of these extra options can be problematic if you need more space or have poor handwriting, but they can help if you struggle to express your work in latex quickly.

    Note that every option results in *a pdf which conforms to our template.* This is crucial. You will not get credit for an assignment that Gradescope rejects because it does not fit the template. You will not get credit for an assignment that is illegible, or has illegible or omitted *name* and *ID* entries.
5. Name rule: your name and ID must be in the specified box at the top of page 1. Replace the `\phantom{...}` command with your name (or ID) so that the code reads something like `\fbox{Kristen Nelson}` to display your name.
6. Figure rule: all figures, graphs, charts, and tables must be labeled correctly. No credit

for figures with unlabeled axes or unlabeled data series.

7. Source-code rule: for problem sets with programming portions you must submit your source code, and it must be runnable. Do not submit Jupyter notebooks as a .ipynb file, submit runnable code in plain-text .py files.
8. Filename rule:
    - For homework/problem sets name your file Lastname-Firstname-MMDD-HX.pdf where Lastname-Firstname are your names separated by a hyphen (if your name(s) consist of multiple words replace spaces with underscores); MMDD is a 2-digit month and 2-digit day for your birthday; HX is "H" for Homework and a number X corresponding to the number of the assignment.
    - Python files must also conform to the above pattern with a py extension instead of pdf. It is a *really* good idea to start your python files with some comment lines specifying your name and student ID.
    - For quizzes the rule is the same as for homework with Q in the filename instead of H.

Advice for assignments

- Read the assignment carefully. There will be instructions (including how to submit) and helpful advice for many questions.
- Many problems may require pen-and-paper thought before you attempt to write the solution up for submission. Being in front of latex code may impede your ability to think creatively and this may be easier with traditional media at first. We expect you to do a lot of rough work that isn't submitted as part of discovering your best, submitted solutions.
- Answer the stated question.
- Justify your answer. Unless the problem specifically says otherwise, every homework problem requires an explanation. Without one, even a perfectly correct solution is worth very little. In particular, saying "it's obvious that" and "trivially" are not explanations.
- Answer the question completely. When a solution requires an algorithm you may need to do the following to gain full credit:
    - Formally restate the problem to make it clear what your algorithm is supposed to achieve.
    - Give concise pseudocode and a description of your algorithm.
    - Explain what the pseudocode does and why, and any indexing or iteration conventions.
    - Prove correctness of the algorithm.
    - Analyze the (e.g. worst-case) running time of your algorithm. Sometimes this can be deduced easily from the structure of the code and sometimes this requires some work solving e.g. a recurrence relation.
    - Your algorithm may need to be suitably clever or efficient to gain credit. Frequently

the problems we study admit a brute-force search algorithm that takes exponential time in the worst case, and we do not consider this mastery of the given problem.

  - Sometime we may demand a particular approach to solving the problem, even if other more efficient solutions exist we want you to think about a certain method for good reasons.

- Write your answers carefully. We have to grade what you write, not what you meant to write. Ambiguous solutions may not receive credit as we cannot infer what you actually know about the problem.
- Use spacing, underlines, boxes, and any reasonable formatting to make your solution structure clear and readable. There are no bonus points for fitting your equations on one line or your entire solution onto one page. In fact this is discouraged when readability is at stake.
- Give generic solutions, not examples. Solutions that include phrases like "and so on" and "etc." typically indicate to us a lack of understanding. Those phrases indicate where you should have used iteration, recursion, or induction precisely but you instead gave a vague description of the task.
- If your algorithm does not work on every valid input then it is not a correct solution. A complete solution explains why there is no input on which your algorithm fails.
- Some programming problems will require you to conduct numerical experiments. For instance, to show that an algorithm takes $O(n \log n)$ time, you will need to measure the number of atomic operations or execution time at multiple values of n, plot the measured values against n, and then plot e.g. the function $n \log n$ showing that the function matches the data reasonably well. Plotting the average time for several runs with a given value of n will almost always improve your results over single times. When presenting such results you must explain your experimental design, and we may suggest helpful tools from the Python standard library to help time or analyze your code.

4 Course policies

Working in teams

- All problem sets must be completed individually.
- We encourage you to form study groups to discuss the problems with each other, and to have discussions on Piazza.
- Each student must independently write up their own solution, the one they submit to demonstrate mastery, for themselves. Working at a whiteboard or on paper together is fine, even if you solve the problem that way, but writing up a joint solution is not.
- As a guideline, write your homework submissions alone, in a separate room from your classmates. Nobody else is allowed to see your submission until after the deadline.
- Quizzes and examinations must be completed individually. Do not discuss the quiz questions until after the deadline.

- Any knowledge or theorem used without proof should be properly cited, e.g. if you use the master theorem state it clearly and tell us where you found the statement (which could be the lecture notes in this case).

Intellectual honesty and plagiarism

- 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 lectures, recitations, or office hours collecting names may be impractical and 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 before the assignment is due.
- Write everything in your own words and cite all outside resources. You are 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 carefully are the lecture notes and the prerequisite material, though it can do no harm to cite them!
- There will be a zero-tolerance approach to violations of this policy. At a minimum you will be permanently barred from being awarded the relevant standard(s) which will seriously affect your final grade. Clear violations will result in removal from the class, being given the grade F, and being reported to the University Honor Council.
- In most editions of this course we catch blatant plagiarism of homework solutions, and this is punished severely as above. You have been warned. The mastery system means that there are plenty of ways to get the grade you deserve and no one assignment or exam is particularly high-stakes, so you are not strongly incentivized to cheat.