Course Syllabus

Jump to Today

INTRODUCTION

This course is about several broad categories of problems and problem-solving techniques (such as greedy algorithms, divide-and-conquer algorithms, dynamic programming, and the class of NP-complete problems) and gaining the tools and experience necessary to judge how a new problem might fit one of these categories, how to approach solving the problem, and how to analyze and adjust your solution.

At the end of the course, you will be able to:

1. Recognize which algorithm design technique(s), such as divide and conquer, prune and search, greedy strategies, or dynamic programming was used in a given algorithm.
2. Select and judge several promising paradigms and/or data structures (possibly slightly modified) for a given problem by analyzing the problem's properties.
3. Implement a solution to a problem using a specified algorithm design paradigm, given sufficient information about the form of that problem's solution.
4. Select, judge and apply promising mathematical techniques (such as asymptotic notations, recurrence relations, amortized analysis and decision trees) to establish reasonably tight upper and lower bounds on the running time of algorithms.
5. Recognize similarities between a new problem and some of the problems they have encountered, and judge whether or not these similarities can be leveraged towards designing an algorithm for the new problem.

You can also see a more detailed list of **topic-level learning goals (https://canvas.ubc.ca/courses/153809/files/34034279)** , although we may alter the specific topics used to address the course-level learning goals.

PREREQUISITES AND CALENDAR ENTRY

Available in the **CPSC 320 UBC Calendar Entry (https://courses.students.ubc.ca/cs/main?pname=subjarea&tname=subjareas&req=3&dept=CPSC&course=320)** .

Beyond these, brush up on probability and combinatorics (e.g., at least the intro to the Wikipedia article on **combinations** **(http://en.wikipedia.org/wiki/Combination)**). You should also be comfortable reasoning using mathematical notation and basic mathematical proof techniques (e.g., direct proof, induction, contradiction). You will need familiarity with summations, sets, relations, functions, asymptotic notation (O, Ω, Θ), recursion, loop invariants, basic data structures (stacks, queues, linked lists, heaps, graphs, hash tables), sorting algorithms, and graph traversal (depth-first and breadth-first search)).

GRADING

Your course mark will be based on the following breakdown. The course staff reserves the right to change this scheme (but does not anticipate using that right).

Assignments (5)	30%
Midterm exams (2)	30%
Final exam	35%
Clicker questions	5%

A small number of students may additionally be rewarded up to 1% bonus on for engagement and high quality contributions on Piazza. To pass the course, you must obtain a 50% overall mark and 45% on the weighted average of your midterm and final exam grades. Students who receive less than 45% on the exam component of their grade will receive as their course grade the minimum of their normally computed grade and 45%.

There will be 5 assignments throughout the term. However, your lowest assignment grade will be dropped.

ASSIGNMENTS

There will be 5 assignments throughout the term. We will drop your lowest assignment mark.

- **Submission***:* Assignments will be submitted by Gradescope. Your *<CWLusername>@student.ubc.ca* email is used to identify you on Gradescope. You can submit your assignment early and you may resubmit as many times as you want before the deadline. **We encourage you to submit something a few days before the first assignment deadline to make sure that everything is working properly.**
- **Typesetting**: Assignments must be written in LaTeX and submitted as a PDF. We will provide the LaTeX source files of the assignments to facilitate this.
- **Late policy***:* We will not accept late assignments or offer any extensions. We already drop your lowest assignment grade, so we do not anticipate waiving a single missed assignment. If you are unable to submit more than one assignment before the deadline, please contact the course coordinator (**cpsc320-admin@ (mailto:cpsc320-admin@cs.ubc.ca) cs.ubc.ca (mailto:cpsc320-admin@cs.ubc.ca)**) and tell us why. We may be willing to consider accommodation. Note that we will ask for supporting documentation in this case.
- **Pseudocode:** We will ask you to write your algorithms in pseudocode -- not actual code. The purpose of pseudocode is to communicate algorithms in such a way that a human being reading your

solution can understand it. We strongly suggest that you accompany your pseudocode with a plain English description of how your algorithm works. We will not run actual code, and will not take your word for it that your algorithm is correct because it succeeded on certain test cases (even if this is true, it doesn't prove that your algorithm is correct!). **You will be penalized if your solution is not communicated clearly enough for the graders to be able to assess correctness just by reading your written submission** (that is, your pseudocode and accompanying explanation). Note: if you want to implement your algorithms in code to test them -- great! This is awesome practice. But that shouldn't be what you hand in to us.

- **Collaboration***:* We allow (and encourage!) you to submit assignments in groups; see the Academic Conduct section below for details.

The assignments are the best learning tool you have in this course. You can learn a fair bit by problem-solving with help during lecture, but completing an assignment problem by yourself (or with a classmate) from start to finish is where you will really develop mastery of the material. As such, we recommend that if you choose to complete the assignments with a group, **every group member should work on every question.** This will tend to result in better assignment quality, and will give you the full learning (and exam preparation) benefit of the assignments. (Not only do different assignment questions target different skills -- so skipping some can leave you with gaps in your skill set -- we often design exam questions that build directly on some assignment questions.) By the same reasoning, we strongly recommend against skipping any assignments, even though we drop your lowest assignment grade.

MIDTERMS

We will have two midterm exams in class on Wednesday October 9 and Wednesday November 6. You **must attend your registered lecture section** for both exams. Both sections will receive different exams. Because of the inherent challenge in ensuring two different exams have equal difficulty, we will apply a multiplicative scaling to the exam scores of the section with the lower median score so that both sections will have the same median grade. (Example: if section A has a median grade of 70% and section B has a median grade of 75%, then the grades of everyone in section A will be multiplied by 75/70 ≈ 1.07, to give both sections a median exam grade of 75%.)

CLICKER QUESTIONS

Clicker questions are graded based on completion — so, if you show up to lecture and answer the question, you will earn a point for it. If `q` is the total number of in-class clicker questions and you earn `y` points total, then your clicker question grade will be `min(1, y/(0.7*q))`. In other words, if you answer 70% of the clicker questions, you will earn full marks for this component. You will need to download **iClicker Cloud** (https://www.iclicker.com/iclicker-canvas-integration/) to your device to be able to participate in clicker questions (see link in the menu on the left).

However, we do not want it to be mandatory to bring a smart device to class every day, so if you would like to opt out of the clicker questions, please email the course coordinator (**cpsc320-admin@ (mailto:cpsc320-admin@cs.ubc.ca) cs.ubc.ca (mailto:cpsc320-admin@cs.ubc.ca)**) no later than September 13 explaining that you would like to opt out, and provide a brief rationale for wanting to do so (don't own a smart phone, prefer not to bring devices to class to avoid digital distraction, etc.). We will transfer the 5% clicker component of your grade to the assignment component (i.e., the assignments will be worth 35%, and all other components will remain the same).

REGRADE REQUESTS

We accept regrade requests for submissions marked on Gradescope (that is, for assignments and the midterm exams) until one week after the submission is returned to you. Your request will be handled by whoever originally graded your submission, possibly with input from other TAs or the instructor. The decision of the grader is final, and may either increase or decrease your grade. Note that we will regrade based solely on what is written in your original submission and not on any additions or clarifications made in your regrade request.

Before you submit a regrade request: please make sure you understand the solution, and check Piazza for a post about common errors and general grading comments. Your regrade request should be clear, concise, and refer specifically to the item(s) on the rubric that you think should or should not have been applied to your submission. (Something like "My friend gave basically the same answer and got a higher grade" is not specific enough.)

What merits a regrade: regrade requests are justified if there was a mistake in applying the rubric. These are the usual circumstances that lead to an increase in points:

- **Your answer is essentially the same as the solution, but the grader didn't realize it.** Make sure your request is clear about why your answer is correct. For example: "I lost points for not initializing my table entries, but I initialized them on line 6," or "My answer is marked as blank, but I wrote my answer on the next page of the exam booklet."
- **Your answer is substantially different from the solution key, but is still correct.** Problems in CPSC 320 often have more than one correct way to solve them. If you had a different approach that you believe is correct, your regrade request should provide a clear, informal explanation of why your approach does in fact work.

What does not merit a regrade: you *should not* submit a regrade request if:

- **You disagree with a judgment call.** For example: "I got 'significant errors' but I think I should have gotten 'minor errors.'" We will not regrade judgment calls. A grader is more likely to make a biased or inconsistent judgment when they're looking at a submission long after the other ones and they've just seen a student's complaint about the grading decision.

- **You disagree with the rubric.** For example: “I lost points for not justifying my answer, but I thought the justification was trivial and didn’t need to be included, so I don’t feel I should be penalized.” You might not like that you lost points under the rubric because of something you did or didn’t do, but we have to grade all submissions consistently. That means we can’t change the grading scheme just for you.

You may be penalized for submitting several bad regrade requests on a single submission (e.g., requests that are poorly explained, reflect poor understanding of the requirements for a correct solution, and/or fall under “what does not merit a regrade”).

For the final exam, you can email the instructor to make an appointment next term to review your exam and learn from it. For marking disputes, see UBC’s **Review of Assigned Standing (http://www.calendar.ubc.ca/Vancouver/index.cfm?tree=3,49,0,0)** policy.

TUTORIALS

Tutorials will begin in week 2 and will generally focus on additional practice with solving problems. On weeks with assignments due tutorials will host office hours instead. You may attend another section besides the one you're registered in. Here is the planned schedule (note that the stat holiday on 10/14 causes a "bump" of a problem-solving tutorial to the following Monday).

Week	***Format***
09/01-09/07	No tutorials
09/08-09/14	Problem-solving tutorial
09/15-09/21	Office hours during tutorials (A1 due)
09/22-09/28	Problem-solving tutorial
09/29-10/05	Office hours (A2 due)
10/06-10/12	Problem-solving tutorial
10/13-10/19	Problem-solving tutorial (Tue,Wed,Fri)
10/20-10/26	Problem-solving tutorial (Mon only), Office hours (Tue,Wed,Fri) (A3 due)

10/27- 11/02	Problem-solving tutorial
11/03- 11/09	Problem-solving tutorial
11/10- 11/16	Office hours (Fri) (A4 due) No tutorials (Mon,Tue,Wed) (Reading Week)
11/17- 11/23	Problem-solving tutorial
11/24- 11/30	Problem-solving tutorial
12/01- 12/07	Office hours (A5 due)

IMPORTANT DATES

Update this for our offering once we have midterm dates.

First day of classes	September 4
Midterm exam 1	October 9 (in class)
Midterm exam 2	November 6 (in class)
Reading week	November 11-13
Last day of classes	December 6
Final exam	TBD (between December 11 and 21)

COMMUNICATION

Course materials will be posted on Canvas. Most other communication will be posted on Piazza. You can register with the link on the Canvas sidebar using your full name (as recognized in the UBC registration system) until September 13. Important announcements will be communicated via Piazza, so we expect you to check the "Announcements" page regularly. Personal communications to the course staff may be posed as private questions on Piazza. For logistical concerns (such as needing to miss a test/exam), please contact the course coordinator at **cpsc320-admin@ (mailto:cpsc320-admin@cs.ubc.ca) cs.ubc.ca (mailto:cpsc320-admin@cs.ubc.ca)** .

PIAZZA GUIDELINES

We will aim to answer most student communications within one business day. Response time may be slower when the course staff is busy (e.g., during exam grading). We will generally not respond on weekends or on holidays. **For publicly posted content-related questions, we will wait at least 4 hours before providing a response**. This is both to encourage independent problem-solving (we often find that students come up with the answer to a question themselves while they wait for a response) and to leave time for peer responses. We will look over peer responses to make sure all questions have been satisfactorily answered, and will leave an endorsement if the peer response is correct and complete, or add responses of our own if we feel anything is missing from peer responses.

To help facilitate learning and keep the course Piazza useful and constructive, we ask that you adhere to the following guidelines:

- **Post content-related questions publicly whenever possible**. If you have a question related to course material (assignments, lectures, etc.), we appreciate your posting these as public (rather than private) questions whenever possible. Doing this allows space for peer responses, and also allows classmates with similar questions to learn from the responses you get. You may post privately to instructors if you choose, but if you do so we will further delay our usual response time and wait at least a day before providing a response. Also, if you do choose to post privately with details of your answer to a question, note that we will not grade answers in advance.
- **Aim to keep your questions clear and concise.** We will try to respond in the same way. Similarly, if you have multiple unrelated questions, it's better to ask them in multiple short posts than in a single long one.
- **Trolling, personal attacks, etc., is cause for removal from the course Piazza.**
- **If you wish to express an opinion (e.g., about challenges in the class, an exam, or issues with students/instructors), communicate it to us face to face, via email, or via private post.** The purpose of the course Piazza page is to facilitate learning, and not to provide a forum for debates about how the course should be run. Any comment/post that disrupts the class, introduces excessive debate, or is (in our opinion) reasonably likely to do either of these things will be removed or switched to only be visible to course staff.

TESTS AND EXAMS

This course has two in-class midterms (on Wednesday October 9 and Wednesday November 6) and final exam (date TBD). The midterm and final exams will be in-person, and you will be allowed to bring cheat sheets (we will provide details on this closer to the test dates).

Missing a test or exam: If you miss one midterm exam due to illness or other extenuating circumstances, you should let the course coordinator know as soon as possible, and no more than 24 hours after the exam (via private Piazza post, or email to **cpsc320-admin@cs.ubc.ca (mailto:cpsc320-admin@cs.ubc.ca)**). We will move the weight of the missed midterm to the final exam. If you miss **both** midterm exams, we will move the weight of the first midterm to the final exam but award a grade of zero for the second midterm. If you are unable to write the **final** due to illness or emergency, contact your Faculty's advising office (e.g., the **Science Undergraduate Advising office (http://www.science.ubc.ca/students/advising/exams)**) immediately.

ACADEMIC CONDUCT

Collaboration enhances the learning experience. We encourage collaboration in various ways throughout the course, subject to the rules stated here:

- **Assignments:** You may work on assignments in groups of up to three people. Groups are described below. Your group may **also** work with any other person or resource subject to five rules:
  - The group must spend at least 15 minutes working on each problem independently before collaborating with others.
  - Collaboration with others must be limited to discussion and brainstorming. No record of **any sort** (e.g., written or electronic material) may be exchanged or leave the brainstorming session. Nor may you bring your solution into the session (except in your head), since that tempts you to annotate it.
  - Each group must write up their own solution independently, using their own words to prove that they understand the problem on their own.
  - Groups must acknowledge all collaborators or sources of assistance in their submission, although you need only name CPSC 320 course staff, handouts, and required textbook if you quote or adapt directly from them. (Despite previous rules, you *may* record the names of people you collaborate with!)
  - We allow the use of ChatGPT and other AI tools only to help you generate LaTeX syntax -- e.g., you are permitted to use ChatGPT to help you produce tikz code to draw a graph as part of your answer to a question. (Though we consider ChatGPT and similar tools to be a "source of assistance", so if you use these to produce any LaTeX syntax, you must acknowledge it in your submission.) You may not use these tools to actually answer the questions (e.g., to help you generate an algorithm, devise a counterexample, or write a proof).
- **Groups: must** submit only one joint solution to the assignment and **will** receive one grade for the assignment that applies to every group member. We urge you to attempt the problems individually for a while, then **come together to meet physically** and hash out a joint solution as a group. Any other approach is likely to lead to disaster on the exams (see Assignment section for more explanation). For advice on group work, speak with the teaching staff and check out **All I Need to Know About Pair Programming I Learned in Kindergarten (http://collaboration.csc.ncsu.edu/laurie/Papers/Kindergarten.PDF)** (lighthearted but well-researched, for programming but applicable to written assignments).
- **Exams** will follow the University's **Rules Governing Formal Examination (http://www.students.ubc.ca/calendar/index.cfm?tree=3,41,90,0)** , including disallowing any

communication by any means with anyone besides the exam's invigilators except where specifically noted in exam instructions.

Violation of any of these rules constitutes academic misconduct and is subject to penalties ranging from a grade of zero on a particular assignment to indefinite suspension from the University. If you are uncertain as to what is or is not reasonable collaboration, or you're having problems understanding or keeping up, please contact the instructor or a TA.

TEXTBOOK

Our required textbook (from which we will have assigned readings!) is: John Kleinberg and Éva Tardos, *Algorithm Design*, Addison-Wesley Publishing company, 2005, ISBN 0-321-29535-8. We have been informed by previous students (but cannot confirm or deny) that the international edition of the textbook is cheaper and yet corresponds closely to the edition listed here.

You may find Kevin Wayne's slides accompanying the book a good supplement: **http://www.cs.princeton.edu/~wayne/kleinberg-tardos/** **(http://www.cs.princeton.edu/~wayne/kleinberg-tardos/)**

(You could even use them for your pre-class readings as long as you're aware that they're not the same as the textbook. "Buyer beware.")

OTHER REFERENCES

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms, 3rd edition*, MIT Press, 2009, ISBN 0-262-03384-4.
- Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, *Algorithms*, McGraw Hill Book Company, 2008, ISBN 0-07-352340-2. (you can find a draft **here** **(http://www.cs.berkeley.edu/~vazirani/algorithms)**).
- Michael Garey and David Johnson, *Computers and Intractability: a Guide to the theory of NP-Completeness*, W.H. Freeman & Company, 1979, ISBN 0-7167-1044-5.
- Donald E. Knuth, *The Art of Computer Programming, Volume 1-4a*.

Online references:

- **Algorithms, etc.** **(http://jeffe.cs.illinois.edu/teaching/algorithms/)** by Jeff Erickson. A good, up-to-date review of algorithms.
- **The Algorithm Design Manual** **(https://www8.cs.umu.se/kurser/TDBAfl/VT06/algorithms/BOOK/BOOK/BOOK.HTM)** by Steven S. Skiena. A somewhat out-of-date, but engagingly written review of algorithms. (Also a print book now in 2nd edition.)

- **Algorithms** **(https://en.wikibooks.org/wiki/Algorithms)** on WikiBooks. (Open.. but spotty and strange in coverage. Maybe you can help!)

Course Summary:

Date	Details	Due