CPSC 320: Intermediate Algorithm Design and Analysis

Home Course Work

Welcome to the Summer 2023 Term 2 edition of CPSC 320! Anne and the entire instructional team is looking forward to getting to know you and supporting you in your learning this summer.

Following are the key resources you need for the course. Click on the Course Work link above to learn what work you will do in this course and how your learning will be evaluated.

Course Staff

Instructor: Anne Condon
Office: ICICS/CS X551
Email: condon at cs.ubc.ca
Office hours: MWF after class

Course Coordinator: Maya Vasdani, cpsc320-admin@cs.ubc.ca
Please email Maya regarding any administrative/logistical issues.

TAs: Information on TAs and their office hours is listed on the course Piazza page, accessible via Canvas.

Learning Goals

In this class you'll learn about useful computational problem-solving techniques, and gain experience both in applying these techniques to solve computational problems efficiently and in recognizing when they are not useful. More specifically, you'll learn how to:

- Come up with clear problem formulations, given poorly specified computational tasks.
- Recognize which algorithm design technique, such as brute force, divide and conquer, greedy, or dynamic programming is used in a given algorithm.
- Evaluate which design technique and/or data structures can be used to solve a given problem efficiently.
- Implement a solution to a problem using a specified algorithm design technique.
- Select and apply mathematical techniques, such as asymptotic notation or recurrence relations, to obtain good upper and lower bounds on the running time of algorithms.
- Recognize similarities between a new problem and problems that you have already encountered, and judge whether or not these similarities can be leveraged to design an algorithm for the new problem.
- Apply the theory of NP-completeness to provide evidence that certain problems may not have algorithms that are efficient on all inputs.

Textbook and Materials

The required textbook is *Algorithm Design* by Jon Kleinberg and Éva Tardos (2005). There will be weekly readings from the textbook, see the Course Work page for details.

Here some other materials that can be useful in your learning:

- Mathematics for Computer Science by Eric Lehman, Tom Leighton and Albert Meyer. Free PDF. This has great background material on proofs, induction, graphs, etc.
- Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., and Stein, Clifford, Introduction to Algorithms, 3rd edition, 2009, ISBN 0-262-03384-4.
- Kevin Wayne's slides based on the Kleinberg-Tardos book.
- Algorithms, etc. by Jeff Erickson.
- 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.

Online Tools

CS Ugrad account:

- If you do not already have an account, it should be set up soon after you register for the class. Associated with this account is your CSID and CS ugrad email address. See here for more information. **Make sure to activate your account right away, or you won't be able to use essential online tools.** Note that your CSID (typically letter-number-letter-number-letter, sometimes letter-number-letter-number) is provided to you when you activate your CS account; it is different from your CWL.

Canvas: You should able to access Canvas within 36 hours of registering for the class. Integrated within canvas you'll find:

- **Piazza**: Use Piazza for communication with peers and course staff about class work, exams, etc. However, please exercise discretion that you do not directly discuss solutions to homeworks that are yet to be submitted. If you have any matter requiring private discussion, you can post privately on Piazza or e-mail the instructor or course coordinator.

- **Gradescope**: Use Gradescope to upload your work on worksheets during lectures, to upload assignments, and to take quizzes and the final exam. Marks will also be on GradeScope. You can access Gradescope from our Canvas course in the left navigation menu.

- Assignments, quizzes, worksheets and tutorials, and solutions to these.

Resources

- **Health & Wellness**. We hope that all students will find this classroom to be a supportive environment, and the course staff are here to help you do your best. Resources are available to help you in times of illness or distress, please check them out and talk with your instructors if we can be helpful.

- **University Policies on Academic Integrity**. All UBC students are expected to behave as honest and responsible members of an academic community. Make sure you understand UBC's academic integrity principles! Breach of UBC's policies with respect to academic honesty can result in serious disciplinary action.

CPSC 320: Intermediate Algorithm Design and Analysis

Home Course Work

Here's roughly what to expect in terms of course schedule, assignment expectations, assessment and so on. Materials other than the textbook will be made available on Canvas. Worksheets for lectures will also be handed out in class.

Lectures and Tutorials

See here for times of lectures and tutorials. Tutorials will start on Wed July 5 (the first week of class) and run twice per week except for the last week, when there will be no tutorial on Fri Aug 11. We expect that you will go to the tutorial for which you are registered. If for some reason that is not possible on occasion, you may attend a different tutorial on that week, but make sure to let your regular tutorial TAs know.

Because you'll learn most by interacting with your peers as well as course staff, lectures and tutorials will involve group and individual problem-solving work. Hard copies of lecture worksheets will be provided in class, and both worksheets and tutorial problems will be provided on canvas in advance of the session where you will need them, and solutions plus annotated worksheets will be posted later.

Here is a tentative course schedule, along with recommended reading.

Week	Topic	Reading
1	Algorithm Design Steps	Chapter 1, pages 1-12 Chapter 2
2	Asymptotic Notation Graph Algorithms	Chapter 2 Chapter 3.1, 3.2, 3.4-3.6
3	Greedy Algorithms Divide and Conquer Algorithms, Recurrences	Chapter 4.1, 4.4-4.7 Chapter 5.1
4	Divide and Conquer Algorithms, Recurrences, Continued Dynamic Programming	Chapter 5.2, 5.3, 5.5 Chapter 6.1
5	Dynamic Programming Continued Reductions	Chapter 6.2, 6.4, 6.5 Chapter 8.1-8.3
6	NP-Completeness	Chapter 8.4, 8.5, 8.6, 8.8

Assignments

Assignments will be available on Wednesdays, including the first week but not the last week, and due at 10pm Vancouver time on Wednesday a week later. So there will be five assignments in total. You can work in groups of size up to three. Each group should make a single submission per assignment.

- The group must spend at least 15 minutes working on each problem independently before collaborating with others.

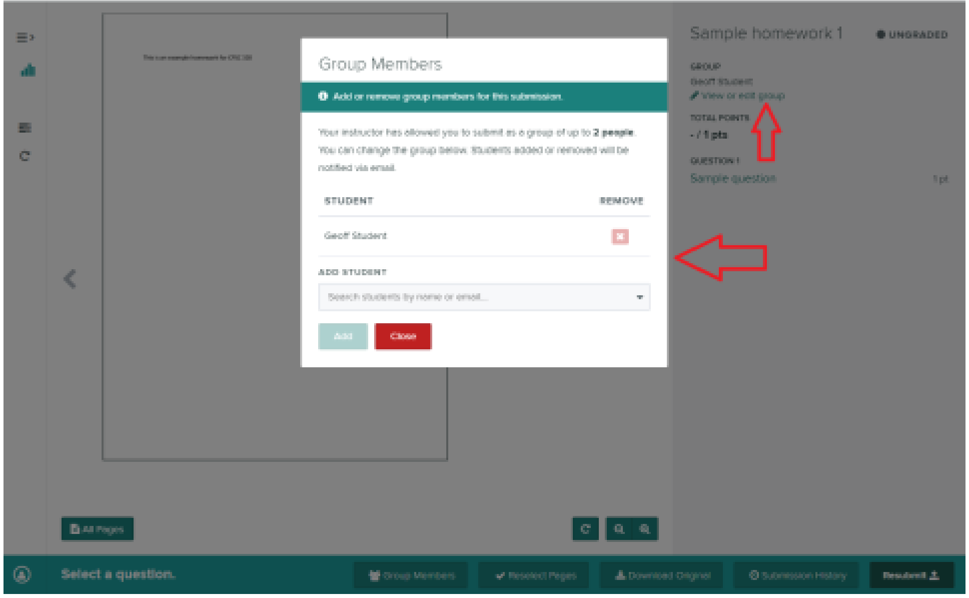
- Collaboration with others is limited to discussion and brainstorming. No record of any sort (e.g., written or electronic material) may be exchanged or leave the brainstorming session.
- After collaborating, each student must take a half-hour break from the problem. Treat electronic resources you consult like written resources (set them aside for a half-hour and then do not consult them while writing your solution).
- Each group must write their own independent solution, using their own words.
- Acknowledge all collaborators or sources of assistance in your submission, although you need only name CPSC 320 course staff, handouts, and textbooks if you quote or adapt directly from them. You should record the names of any other people you collaborate with.

We will make the .tex version of the assignment available. You should prepare your solutions using latex. We will accept solutions prepared using other good formatting systems, as long as they are clearly legibile. Solutions that are handwritten or difficult to read will receive a grade of zero.

Submit your assignment **in pdf format** on Gradescope by 10pm Vancouver time. You can submit early and resubmit as often as you want up to the deadline. **We strongly encourage you to submit something at least a day in advance of the first assignment**, to make sure that things are working properly.

After uploading to Gradescope, link each question with all the pages of your pdf containing your solution.

Also, add names of group members (as recorded on Canvas) on GradeScope *after* one student has made the initial submission. See the image below for adding group members to a submission:



Group Members

Add or remove group members for this submission.

Your instructor has allowed you to submit as a group of up to **2 people**. You can change the group below. Students added or removed will be notified via email.

STUDENT	REMOVE
Geoff Student	☒

ADD STUDENT

Search students by name or email...

As a secondary failsafe, please clearly write the names of all group members on the first page of each submission, and if you want an extra double-check on your identity, include your student number.

Late homework submissions will be accepted up to 24 hours past the due date, at a penalty of 15% of the assignment's full value. To avoid the deduction, have your documents imaged and ready to upload ahead of time, to avoid any last-minute stress and glitches. Remember that you can always upload a draft ahead of time, and resubmit later.

To manage TA workload, we will select some, but not all, parts of the assignment for grading.

Quizzes

Instead of midterms, there will be a weekly quiz during lecture, **at 11:00am on Fridays**, except for the first and last weeks. There will be four quizzes in total. You will have 50 minutes to do the quiz.

The quiz will test topics relating to reading, lectures, assignment and tutorials from the previous Friday up to Wednesday of that week, including reading for that week. Material from earlier weeks, or course prerequisites, may also appear on a quiz.

Do not write a quiz if you feel ill. If you are unable to write a quiz, inform the course coordinator (cpsc320-admin@cs.ubc.ca) immediately.

Your lowest mark in a quiz will be dropped. If you miss two quizes, your mark for one will be dropped and your mark for the second will be calculated based on your performance on the final. You will not be able to make up marks for additional quizzes should you miss more than two.

Final

We will post the date of the final exam here once we know it. The material covered in the final exam will be cumulative over the whole course. If you are unable to write the final examination due to illness, you must apply to your home faculty for Academic Concession as soon as possible.

Evaluation

Five assignments × 5% = 25%
Three quizzes × 10% = 30% (there will be four quizzes, but lowest grade will be dropped)
One final exam × 45%

In accordance with UBC Policy #65, students who are scheduled to write examinations or attend tutorials on the holy days of their religion must notify their instructors two weeks in advance of the religious holiday they wish to observe. Instructors will provide opportunity for students to make up the missed work without penalty.

To pass the course, you must obtain at least 50% on the weighted average of the quizzes and final exam. Students who fail to meet this requirement will receive at most a mark of 45% for the course, regardless of the results of the formula above.

Regrading Requests

We do our best to grade all submissions fairly and consistently. Because graders occasionally make mistakes, we welcome regrade requests that help us correct such mistakes and ensure fairness. You have one week from the time that the grades are posted to submit a regrade request on Gradescope. Please be respectful and help us ensure that time spent on regrades is productive by following these guidelines.

- Prior to making a regrade request, make sure you understand the solution to the problem, and read any comments provided on Gradescope or Piazza about how the problem was graded. If you don't understand what's missing from your answer, come see one of us during office hours before submitting a regrade request.
- Explain clearly and specifically why you believe your mark is inappropriate, referring to your solution, and the rubric. Good examples: "I lost a point for not initializing my table entries, but I initialize them in line 6." Or, "My solution is quite different from the one in the sample solution, but I'm confident that my approach also works for the following reason..."
- Don't submit a request because you think the rubric is unfair, or if you disagree with a judgment call of the grader. A bad example that disputes the rubric is: "I lost a mark for not justifying my answer, but I didn't think I needed to justify my answer, and I don't feel I should be penalized." A bad example that disputes a judgment is: "I should get 'minor errors (-0.5 points)' instead of 'significant errors (-2 points)'." An example that is not sufficiently specific is: "My friend gave basically the same answer and got a higher grade."

The person who graded the question will review your request, possibly with input from an instructor or other TAs. The decision of your TA or instructor is final. It may either increase or decrease your mark. Submitting several poorly-explained regrade requests may result in a penalty.