CSC263H1 Y LEC5101 20225:Data Structures and Analysis

Jump to Today ✎ Edit

Welcome to CSC263!

Please start by reading the entire syllabus (this page) and then checking the course Modules (use the link on the left). The material posted on Quercus is required reading. You are responsible for all announcements made in lecture and on Quercus.

This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TAs, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com.

Find our class signup link **here** (https://piazza.com/utoronto.ca/summer2022/csc263h1ylec5101) .

To contact me regarding *private/personal/administrative issues* related to CSC263, please use the course email address: **csc263-2022-05@cs.toronto.edu** (mailto:csc263-2022-05@cs.toronto.edu)

Lectures and Tutorials

Times, instructor names, and locations for lectures and tutorials.

Section	LEC 5101	TUT 5101	TUT 5201	TUT 5301	TUT 5401
Time	Wed 18:00-20:00	Wed 20:00-21:00	Wed 20:00-21:00	Wed 20:00-21:00	Wed 20:00-21:00
Instructor	Sean Ovens	TBA	TBA	TBA	TBA
Location	KP 108	BA 2165	BA 2195	BA 2159	BA 2139

- Recordings of all lectures, slides, and worksheets for all tutorials will be posted to the **Lectures** page.
- Tutorials give students the chance to apply the material they learned during lecture by completing extra (not for marks) worksheets. Tutorials are also an excellent opportunity to ask TAs about course content or problem sets.

Office Hours

There will be two office hours every week, beginning on May 17. Office hours will be held on Tuesdays 13:00-14:00 and Wednesdays 13:00-14:00 in BA 4290.

Learning Outcomes

During this course, students will become familiar with several standard data structures and abstract data types. These include graphs, dictionaries, AVL trees, hash tables, heaps, disjoint sets, and more. Students will also learn about standard runtime complexity measures (i.e. best/worst-case, average-case, and amortized complexity). More precisely, they will be able to:

1. recognize algorithms that employ each data structure,
2. write algorithms that employ each data structure,
3. recognize when each complexity measure is most appropriate,
4. analyze the efficiency of algorithms using each complexity measure, and
5. choose/modify data structures appropriately to solve various problems.

Marking Scheme

Description of all assessments, their weights, and whether they are to be completed individually or in a group.

Assessment (quantity)	Weight	Description	Individual/group
Weekly Quercus modules (10)	10%	Each worth 1%	Individual
Problem sets (3)	45%	Each worth 15%	Individual/pair
Midterm (1)	15%	Completed during tutorial time slot	Individual
Final exam (1)	30%	Must get at least 40% on final exam to earn more than a D in the course	Individual

---

Quercus Modules

Starting in Week 2, you will complete a weekly Quercus Module worth 1%, due by 18:00 every Wednesday. These modules must be completed **individually** (without partners), and will have components marked for correctness. You may submit answers as many times as you wish (up to the deadline), but **only your last on-time submission will be marked**. Each Module will consist of a combination of the following elements.

- **Demonstrate**: Quiz questions that give the opportunity to demonstrate and exercise the main concepts from the previous week's lectures and tutorial.
- **Discover**: Readings or links to a video or simulation where new material is introduced. CSC263H1 is not completely “flipped”, unlike courses like CSC108H1 and CSC209H1. However, some of the easier concepts will be taught through Discover components. You must complete these components before the following lectures. This allows the lectures to go further by building on the content of the Discover modules, instead of having to “waste” lecture time going over the easiest concepts. Each Discover component will usually be paired with a Describe component.
- **Describe**: Short quiz questions about new material from an associated Discover component. If you find that you cannot answer these questions, you should go back and redo the Discover activity more carefully, before trying the Describe quiz again. You may also find it helpful to consult the relevant chapters in the course textbook, for additional explanations and examples.
- **Review**: Short quizzes that mostly test prerequisite material (concepts that you are expected to know from previous courses). If you are not confident about your answers to a review quiz, please take the time to review the corresponding material from your prerequisite courses and then retake the quiz (before its deadline, of course).

*Please aim to complete the Quercus Modules **well before** their submission deadline (18:00 on Wednesdays).* **Late submissions will NOT be accepted under any circumstance**. Each module will be available well before it is due, so you have plenty of time to attempt it before Wednesday and to work around any technical difficulties you may encounter. If you face an unexpected personal emergency that makes you incapable of doing any coursework *for multiple days before the due date,* please request special consideration.

Problem Sets

- All problem sets will be submitted electronically on MarkUs.
- Your submissions must be fully typeset and we will only accept .pdf files.
- You may work individually or with one partner. If you are working as a pair, you must declare your partner on MarkUs well before the due date. For assistance with MarkUs, see the following **Documentation for Students (https://github.com/MarkUsProject/Markus/wiki/Student-Guide#how-to-form-group)**. Only one student from each pair should create the group and invite the other using their UTORid.
- We recognize that unexpected problems sometimes make it difficult to submit problem sets on time. For this reason, we will accept late problem sets with a penalty. There is a one hour grace period after the due date in which no late penalty is applied. Following this, the penalty increases by 5% of the total possible mark per hour. For example, if a problem set that is out of 20 total marks is turned in between 1 and 2 hours late, then the final mark for this submission will be reduced by 0.05*20 = 1 mark. Problem sets that are turned in more than 10 hours late will not be marked (unless there is some emergency or other exceptional circumstance - see the Special Consideration section below).

Deductions for late submissions.

Time of submission	Penalty
on time or early	no penalty
less than 1 hour late	no penalty
less than 2 hours late	5%
less than 3 hours late	10%
less than 4 hours late	15%
less than 5 hours late	20%
less than 6 hours late	25%
less than 7 hours late	30%
less than 8 hours late	35%
less than 9 hours late	40%
less than 10 hours late	45%
10 hours late or later	100% (not marked)

Tests and Final Exam

There will be one (1 hour) midterm at the start of lecture on July 13. The final exam will be scheduled by the Faculty of Arts & Science. All assessments will be in person.

Reporting a Marking Error

If you believe there was an error in the marking of an assignment, you may report that error in MarkUs. Only error reports submitted within the time frame indicated on MarkUs will be accepted - this is typically within two weeks of when feedback for the assessment was returned. Along with your request, you must clearly indicate the portion of the assignment you wish to be reassessed, as well as the reason you believe your original mark is incorrect. Your grade may increase, decrease, or remain the same after your request has been processed.

Academic Integrity

All of the work you submit must be done by you (or your problem-set partner), and your work must not be submitted by someone else. Plagiarism is academic fraud and is taken very seriously. Please read the Rules and Regulations from the U of T Governing Council (especially the Code of Behaviour on

Academic Matters): **http://www.governingcouncil.utoronto.ca/policies/behaveac.htm (http://www.governingcouncil.utoronto.ca/policies/behaveac.htm)**

Please also see the information for students from the Office of Student Academic Integrity: **https://www.artsci.utoronto.ca/current/academic-advising-and-support/student-academic-integrity (https://www.artsci.utoronto.ca/current/academic-advising-and-support/student-academic-integrity)**

Accessibility Needs

The University of Toronto is committed to accessibility. If you require accommodations or have any accessibility concerns, please visit **http://www.accessibility.utoronto.ca (http://www.accessibility.utoronto.ca/)** as soon as possible.

Special Consideration

Students experiencing illness or other emergencies that prevent them from being able to complete homework on time, or write tests, can apply to the instructors for special consideration. You will be required to affirm that you are abiding by the **Code of Behaviour on Academic Matters (http://www.governingcouncil.utoronto.ca/policies/behaveac.htm)**, in particular that it is an offence

> *to engage in any form of cheating, academic dishonesty or misconduct, fraud or misrepresentation not herein otherwise described, in order to obtain academic credit or other academic advantage of any kind*

That is, that you are truly experiencing an emergency, and acknowledge that to falsely claim so is an academic offence. Applying does not guarantee that you will be granted special consideration.

To apply for special consideration in CSC263, complete the **Request for Special Consideration Form** and email it to the course account (**csc263-2022-05@cs.toronto.edu (mailto:csc263-2022-05@cs.toronto.edu)**) *from your UofT email address*. You will receive an email response to your request within 1-2 business days.

IMPORTANT: *Submit your request soon as possible if you find yourself in such a situation. It is easier to resolve situations earlier rather than later.* If your emergency will affect your ability to complete coursework for more than a few days, or in multiple courses, we recommend you also talk to your registrar. You should also complete the absence declaration form on ACORN.

Video Recording and Sharing Policy

This course, including your participation, will be recorded on video and will be available to students in the course for viewing remotely and after each session. Course videos and materials belong to your instructor, the University, and/or other source depending on the specific facts of each situation, and are protected by copyright. In this course, you are permitted to download session videos and materials for your own academic use, but you should not copy, share, or use them for any other purpose without the explicit permission of the instructor. For questions about recording and use of videos in which you appear please contact your instructor.

Textbook

The (optional) textbook, **Cormen, Lieserson, Rivest & Stein: Introduction to Algorithms, *3rd edition* (https://mitpress.mit.edu/books/introduction-algorithms-third-edition)** is available online from the University of Toronto library.

Course Summary:

Date	Details	Due
Tue Jan 18, 2022	**Demonstrate: Week 1 Concepts and Reading the Syllabus** (https://q.utoronto.ca/courses/263250/assignments/856394)	due by 12pm
	Review: Binary Trees (https://q.utoronto.ca/courses/263250/assignments/856393)	due by 12pm
Tue Jan 25, 2022	**Describe: Dictionaries** (https://q.utoronto.ca/courses/263250/assignments/856446)	due by 12pm
	Review: Binary Search Trees (https://q.utoronto.ca/courses/263250/assignments/856448)	due by 12pm
Wed May 18, 2022	**Demonstrate: Binary Search Trees** (https://q.utoronto.ca/courses/263250/assignments/856172)	due by 6pm
	Describe: AVL Trees (https://q.utoronto.ca/courses/263250/assignments/856171)	due by 6pm
Wed May 25, 2022	**Demonstrate: AVL Trees** (https://q.utoronto.ca/courses/263250/assignments/856173)	due by 6pm

Date	Details	Due
	Describe: Hashing (https://q.utoronto.ca/courses/263250/assignments/856174)	due by 6pm
Wed Jun 1, 2022	Demonstrate: Augmentation (https://q.utoronto.ca/courses/263250/assignments/856444)	due by 6pm
	Demonstrate: Hashing (https://q.utoronto.ca/courses/263250/assignments/856442)	due by 6pm
	Describe: Priority Queue ADT (https://q.utoronto.ca/courses/263250/assignments/856392)	due by 6pm
Wed Jun 8, 2022	Demonstrate: Heap Operations (https://q.utoronto.ca/courses/263250/assignments/856447)	due by 6pm
	Describe: Quicksort (https://q.utoronto.ca/courses/263250/assignments/856443)	due by 6pm
Wed Jun 15, 2022	Demonstrate: Randomized Quicksort (https://q.utoronto.ca/courses/263250/assignments/856460)	due by 6pm
	Describe: Dynamic Arrays (https://q.utoronto.ca/courses/263250/assignments/856459)	due by 6pm
Wed Jul 6, 2022	Demonstrate: Amortized Analysis (https://q.utoronto.ca/courses/263250/assignments/856463)	due by 6pm
	Describe: Graphs (https://q.utoronto.ca/courses/263250/assignments/856462)	due by 6pm
Mon Jul 11, 2022	Demonstrate: Graph Representations and BFS (https://q.utoronto.ca/courses/263250/assignments/856469)	due by 12pm
Wed Jul 20, 2022	Demonstrate: Depth First Search (https://q.utoronto.ca/courses/263250/assignments/856608)	due by 6pm

Date	Details	Due
	Describe: Spanning Trees (https://q.utoronto.ca/courses/263250/assignments/856609)	due by 6pm
Wed Jul 27, 2022	**Demonstrate: Minimum Spanning Trees** (https://q.utoronto.ca/courses/263250/assignments/856612)	due by 6pm
	Demonstrate: Strongly Connected Components (https://q.utoronto.ca/courses/263250/assignments/856611)	due by 6pm
	Describe: Disjoint Set ADT (https://q.utoronto.ca/courses/263250/assignments/856610)	due by 6pm
Wed Aug 3, 2022	**Demonstrate: Disjoint Sets** (https://q.utoronto.ca/courses/263250/assignments/856614)	due by 6pm
	Describe: Comparison Trees (https://q.utoronto.ca/courses/263250/assignments/856613)	due by 6pm