Navigation	**Home page (https://q.utoronto.ca/courses/395904/pages/csc373-fall-2024-algorithm-design-analysis-and-complexity)**	**Lectures (https://q.utoronto.ca/courses/395904/pa**

Overview

Welcome to the course webpage for the Fall 2025 term of CSC373 Algorithm Design, Analysis, and Complexity. Here is the course content description:

> Standard algorithm design techniques: divide-and-conquer, greedy strategies, dynamic programming, linear programming, randomization, network flows, approximation algorithms. Brief introduction to NP-completeness: polynomial time reductions, examples of various NP-complete problems, self-reducibility. Additional topics may include approximation and randomized algorithms. Students will be expected to show good design principles and adequate skills at reasoning about the correctness and complexity of algorithms.

All course announcements will be posted here on Quercus. Check the **Lectures (https://q.utoronto.ca/courses/395904/pages/lectures)** page for the required reading.

Where and When

There are two sections for this course. Make sure you look up the location and time for the section you are registered for.

LEC0101

Time	Lecture	Type
Mon 11am-12pm	KP 108	Lecture
Wed 11am-1pm	MP 203	Lecture
Fri 12pm-1pm	AB 107	Tutorial (TUT 0101)
Fri 12pm-1pm	ES B149	Tutorial (TUT 0102)

LEC0201

Time	Lecture	Type
Mon 3pm-4pm	BA 1190	Lecture
Wed 3pm-5pm	KP 108	Lecture
Fri 3pm-4pm	BA 1200	Tutorial (TUT 0201)
Fri 3pm-4pm	BA 1210	Tutorial (TUT 0202)

Contact information

Instructor	Aleksandar Nikolov
Course Email	**csc373-2025-09@cs.toronto.edu (mailto:csc373-2024-09@cs.toronto.edu)**
Office	SF 3316
Office Hours Location:	TBA
Office Hours:	Tuesday 11am-1pm

Prof. Nikolov will attempt to respond to legitimate email inquiries from students within 48 hours. **Please use the course email for all course related matters.** Your email will go to a ticketing system and will be answered promptly by the course team or by instructional support staff. Please send me an email to make an appointment before stopping by my office, *unless you are coming during the regularly scheduled office hours.*

Textbook(s)

Reading assignments will be given from these sources:

- (*Required*) Introduction to Algorithms by Cormen, Leiserson, Rivest, and Stein [CLRS]
- (*Recommended*) Algorithm Design by Kleinberg and Tardos [KT]

- (*Recommended*) Algorithms by Jeff Erickson [E]

The 3rd edition of CLRS (which I use as well) is available online from the **U of T Libraries (https://librarysearch.library.utoronto.ca/permalink/01UTORONTO_INST/14bjeso/alma991106991137506196)** . An ebook version of KT is available for purchase from the **U of T Bookstore (https://www.uoftbookstore.com/adoption-search)** . E is available **online (https://jeffe.cs.illinois.edu/teaching/algorithms/)** . More information about assigned readings is posted in the **Lectures (https://q.utoronto.ca/courses/395904/pages/lectures)** page.

Piazza

You can access Piazza from within Quercus. When using Piazza, be respectful to your instructors and fellow students. Offensive language and threatening behavior will not be tolerated. Keep in mind that when posting "anonymously", you are anonymous only to other students, but not to the instructors.

Grading Scheme

Your mark for the class will be based on the following components:

- 6 Quizzes: 3% (0.5% each)
- 4 Homework assignments: 24% (6% each)
- Embedded ethiCS module: 5%
- 2 term tests: 30% (20% term test 1, 10% term test 2)
- Final Exam: 38%

Quizzes will be roughly bi-weekly, to be complete individually on Quercus. Homework assignments will be done in groups of at most two. See **Homeworks (https://q.utoronto.ca/courses/395904/pages/homeworks)** for more information.

There will be *term tests*. They are scheduled as follows:

- Term test #1: October 22, during your regularly scheduled lecture time slot, 110 mins long;
- Term test #2: November 28, during your regularly scheduled tutorial time slot, 50 mins long.

Test test #1 will cover roughly the first six weeks of the course (not including Network Flows but everything before that), and term test #2 will cover weeks 7-10 (Network Flows, Linear Programming, and Computational Complexity).

There will be a comprehensive *final exam*. **You need to score at least 40% on the final exam to pass the course.**

Special Consideration

Lectures

Navigation	**Home page (https://q.utoronto.ca/courses/395904/pages/csc373-fall-2024-algorithm-design-analysis-and-complexity)**	**Lectures (https://q.utoronto.ca/courses/395904/pa**

General Info

The **lectures** allow us to explain new material, how it relates to the rest of the course (and what you've learned in other courses), and to show examples of applying the material. You are expected to read the assigned material, and attempt the exercises in the reading on your own time. It is recommended that you make your own notes during lectures and tutorials. Slides and recordings from the lectures will be posted on this page. Nevertheless, you are strongly encouraged to attend lectures, so that you can participate in class discussions and collective problem solving.

Tutorials are an opportunity to apply what you have learned in lectures by solving algorithm design problems. One or more problems will be posted on the course webpage before the tutorial, and students are expected to study the problem and attempt to solve it prior to the tutorial. During the tutorial you will be able to discuss your solution with your classmates, and your TA will explain the solution. Solutions will typically be posted some time after the tutorial. Solving the problems in tutorials will allow you to deepen your understanding of the material, and prepare for homework assignments and tests. Problems from the tutorials will often be similar to problems you will encounter on assignments and tests.

Students often learn a lot from working with one another. You are encouraged to meet with other students taking the course. For example, you might work through exercises in the course text together or discuss any material you found confusing in lecture or in the text.

Tentative Schedule of Lectures

Readings from the 3rd edition of Introduction to Algorithms, by Cormen, Leiserson, Rivest, and Stein are marked by CLRS. You can find an **online version (https://librarysearch.library.utoronto.ca/permalink/01UTORONTO_INST/14bjeso/alma991106378851006196)** of the 3rd edition from the U of T Libraries website. Note that chapter and section numbers may differ in different editions.

Readings from Algorithm Design by Kleinberg and Tardos will be marked by KT. You can purchase an ebook version from the **U of T Bookstore (https://www.uoftbookstore.com/adoption-search)** .

Readings from Algorithms by Jeff Erickson will be marked by E. The book is available **online (https://jeffe.cs.illinois.edu/teaching/algorithms/)** .

Week and Topic	Readings	Tutorials
Week 1: Sept 1–7 Divide & Conquer Algorithms	CLRS Ch 4; KT Ch 5; E Ch 1 **Intro Slides (https://q.utoronto.ca/courses/395904/files/38959704?wrap=1)** ⤓ **(https://q.utoronto.ca/courses/395904/files/38959704/download?download_frd=1)** **Divide and Conquer Slides (https://q.utoronto.ca/courses/395904/files/38959998?wrap=1)** ⤓ **(https://q.utoronto.ca/courses/395904/files/38959998/download?download_frd=1)** **D&C Annotated Slides (https://q.utoronto.ca/courses/395904/files/38997606?wrap=1)** ⤓ **(https://q.utoronto.ca/courses/395904/files/38997606/download?download_frd=1)**	**D&C Exercises (https://q.utoronto.ca/cou** ⤓ **(https://q.utoronto.ca/cou download_frd=1)**
Weeks 2 & 3: Sept 8–21 Dynamic Programming	CLRS Ch 15, 24.1, 24.5; KT Ch 6; E Ch 3	**DP Exercises (https://q.utoronto.ca/cou** ⤓ **(https://q.utoronto.ca/cou download_frd=1)** **DP Exercises-p2 (https://q.utoronto.ca/cou** ⤓ **(https://q.utoronto.ca/cou download_frd=1)**
Weeks 4 & 5: Sept 22-Oct 5 Greedy Algorithms	CLRS Ch 16, 24.3; KT Ch 4; E Ch 4	**Greedy Exercise (https://q.utoronto.ca/cou** ⤓ **(https://q.utoronto.ca/cou download_frd=1)**

		Greedy Exercise 2 (https://q.utoronto.ca/cou ⤓ (https://q.utoronto.ca/cou download_frd=1)
Weeks 6 & 7: Oct 6–19 Network Flow Algorithms	CLRS Ch 26; KT Ch 7; E Ch 10	Network Flows Exerci (https://q.utoronto.ca/cou ⤓ (https://q.utoronto.ca/cou download_frd=1)
Week 8: Oct 20–26 Network Flow Applications **Term test 1 on Oct 22**	CLRS Ch 26; KT Ch 7; E Ch 11	Network Flow Exercise (https://q.utoronto.ca/cou ⤓ (https://q.utoronto.ca/cou download_frd=1)
Fall break: Oct 27-Nov 2		
Nov 5: Embedded Ethics Module	TBA	
Weeks 9 & 10: Nov 3–16 Linear Programming	CLRS Ch 29.1, Ch 29.2 and 29.4 (without the proofs) LP Examples Chapter (https://q.utoronto.ca/courses/395904/files/38324090?wrap=1) ⤓ (https://q.utoronto.ca/courses/395904/files/38324090/download?download_frd=1)	Linear Programming E (https://q.utoronto.ca/cou ⤓ (https://q.utoronto.ca/cou download_frd=1) LP Exercises 2 (https://q.utoronto.ca/cou ⤓ (https://q.utoronto.ca/cou download_frd=1)
Weeks 11 & 12: Nov 17-30	CLRS Ch 34; KT Ch 8; E Ch 12	NP Completeness Exe (https://q.utoronto.ca/cou

Computational Complexity **Term Test 2 on Nov 28**		⤓ **(https://q.utoronto.ca/cou** **download_frd=1)**
Dec 1-2 Approximation Algorithms	CLRS Ch 35; KT Ch 11	

Learning Objectives

By the end of this course, you should have the following skills

- Know the three basic algorithm design paradigms - divide and conquer, greedy, dynamic programming. Be able to recognize when an algorithm uses one of the paradigms, when a paradigm is the appropriate one for a given problem, and apply the paradigms to design new algorithms.
- Be able to model problems using network flow and linear programming, and recognize which problems can be modeled in this way.
- Be familiar with basic algorithms for solving network flow problems, in particular the Ford-Fulkerson algorithm.
- Know the definitions of the complexity classes P and NP, understand the significance of the P vs NP problem, understand the concepts of NP hardness and NP completeness. Be able to use reductions to prove NP-hardness.

Homeworks

Navigation	**Home page** (https://q.utoronto.ca/courses/395904/pages/csc373-fall-2024-algorithm-design-analysis-and-complexity)	**Lectures** (https://q.utoronto.ca/courses/395904/pa

Overview

The course has 4 *group assignments* to be completed in groups of maximum two people each, and submitted on MarkUs. Additionally, a short quiz will be released on Quercus roughly every other week on Monday, and due by the end of the day the Friday of the same week. Quizzes should be completed *individually.* Quizzes check basic understanding of the material covered in the lectures and the assigned readings, while group assignments check that you can use the new concepts and techniques creatively to solve algorithmic problems.

Quizzes will generally be posted on Monday after lecture, every other week. They will be due by noon the following Monday. Each group assignment is due **by midnight** on its due date, see below for the schedule.

Assignment	Date out	Date due	Problems/Solutions
Assignment 1	Sept 12	Sept 26	
Assignment 2	Sept 26	Oct 10	
Assignment 3	Oct 17	Nov 4	
Ethics Post-Module Assignment	Nov 5	Nov 10	
Assignment 4	Nov 14	Dec 1	

Additionally, we will have an **Embedded Ethics Module**, with several associated assignments:

- *pre-module survey*, released on Sept 23, due on Oct 1 at 3pm, worth 0.5% of your grade;
- *post-module survey*, released on Nov 29, due on Dec 2 at 11:59pm, worth 0.5% of your grade;
- *pre-module assignment*, which will be submitted together with Assignment 3, and will consist of one additional homework question, worth 2% of your grade;
- *post-module assignment*: short essay questions related to the module, released on Nov 6, due on Nov 11 at 11:59pm, worth 2% of your grade.

Working in Groups

A *group assignment* is to be done by at most two people. For these assignments you are strongly encouraged to work with a partner, rather than work alone. You and your partner should discuss the questions with one another, and come up with solutions *together*, but you may **not** discuss them with other students. The first page of your submission **must** list the name, student ID, and UTOR email address of each group member.

If you would like to work with someone but you don't know anybody who could be your partner, simply post a "Search for Teammates" message in Piazza. Also, make an effort to speak with your classmates during lectures and tutorials - you may find that there are many others in the same situation as yours.

Academic Integrity

When working on assignments, you are not allowed to use other resources except the assigned reading materials, and the materials posted on this site. You should **not** discuss homework solutions with anyone other than the professor, the TA, and, if working on a group assignment, your partner.

Failure to comply with these guidelines is a serious academic offense.
If you have any questions about this policy, make sure you ask the professor or the TA. More information about why plagiarism is bad and what happens to cheaters can be found at **http://www.cs.toronto.edu/~fpitt/documents/plagiarism.html** **(http://www.cs.toronto.edu/~fpitt/documents/plagiarism.html)** .

Lateness Policy

Every student has **three** grace credits, and each credit allows you or your group to be late on one *group assignment* for up to 24 hours. You can use up to two credits on one assignment. After the credits are used, no other late submission from the same student will be accepted for the remainder of the course (except in special circumstances after submitting the special considerations form below). If you are working in a group, then the credit is taken from both members of the group. *Grace credits cannot be combined with other extensions.*

Special considerations form coming soon. See the **Home** **(https://q.utoronto.ca/courses/395904/pages/csc373-fall-2024-algorithm-design-analysis-and-complexity)** page for more details on special considerations.

Remarking Requests

Remarking requests will be accepted up to one week after the date a homework assignment is returned, and should be submitted on MarkUs. A remarking request can be used to alert us to possible mistakes in the grading of an assignment, but *not to question the marking scheme of the assignment.* In response to the remarking request your mark may go up or down.

Submission Instructions

Quizzes will be posted and submitted here on Quercus.

Group assignment submissions will be done using MarkUs. All group assignments should be **typed and not handwritten.**

To submit as a group (**only for group assignments**), one of you needs to *invite* the other to be their partner, and then the other student needs to accept the invitation. To invite a partner, navigate to the appropriate Assignment page, find "Group Information", and click on "Invite". You will be prompted for the other student's username. To accept an invitation, find "Group Information" on the assignment page, find the invitation listed there, and click on "Join". You should do this **before the deadline** even if you are planning to use your grace credit or have been granted an extension.

Once you have submitted, click on the file's name to check that you have submitted the correct version-**and that it is in PDF**.

Remember to put the name, student ID, and UTOR email address of the group member who wrote the solution, and also the name, student ID, and the UTOR email address of the group member who proof-read and revised it on the first page of the submission.

File Formats and LaTeX resources

You are encouraged to use LaTeX to typeset your homework solutions (see below for links to LaTeX resources). However, the use of LaTeX is not required - what matters is that your submissions all be in PDF and typed. **Scans of hand-written solutions will not be accepted!**

LaTeX resources LaTeX is a general-purpose typesetting system that makes it easy to generate high-quality documents, particularly when formatting mathematical formulae. In addition, Piazza supports typesetting equations with LaTeX syntax (by enclosing the equation in double dollar signs, for example `$$e^{2\pi i} - 1 = 0$$`). Here are some links to get you started.

- **TeXworks** (http://www.tug.org/texworks/), a cross-platform LaTeX front-end.
- The **LaTeX Wikibook** (http://en.wikibooks.org/wiki/LaTeX).
- Additional **LaTeX Documentation** (http://www.latex-project.org/guides/), from the home page of **the LaTeX Project** (http://www.latex-project.org/).
- **Overleaf** (https://www.overleaf.com/) allows you to create and collaboratively edit a LaTeX without having to install LaTeX on your machine.

Special consideration form coming soon! We will aim to then respond to you within two work days. *Please do not send an email in addition to submitting the form: the form has all the information we need.*

Please make sure to request special consideration as soon as possible, and definitely **before** the posted due date (for assignments) or the test time (for term tests).

If you have an accommodation letter from Accessibility Services that entitles you to extra time for homework assignments, and you plan to use this accommodation, please fill out the form above, and attach the letter. Please do so each time you need to use the accommodation.

Academic Integrity

Every student must abide by the **University of Toronto academic integrity policy (https://www.academicintegrity.utoronto.ca/)**, and the **Code of Student Conduct (http://www.governingcouncil.utoronto.ca/Assets/Governing+Council+Digital+Assets/Policies/PDF/ppjul012002.pdf)**. Academic misconduct is taken **very seriously**! See the **Homeworks (https://q.utoronto.ca/courses/395904/pages/homeworks)** page for information about what resources you are allowed to use when working on your assignments.

AI Assistants Policy: Think of AI assistants (i.e., LLM chat bots and similar) as another student in the course. You can discuss the material with them, ask questions, ask for help on practice problems, ask about topics that go beyond the scope of the course. However, you can only partner for group homework assignments with one other registered student in the course. Just as you are not allowed to ask a student who is not your partner for homework assignment problem solutions, you **should also not ask an AI assistant to complete your assignments for you.** This is just as inappropriate as asking another student. Also keep in mind that you can 100% trust the information you get from an AI assistant, and you should always double check it. Asking me or the TAs for information about the course and the material is still the most reliable way to get help.