# ECE 606, Fall 2021, Syllabus, Logistics and Schedule

Algorithms are at the very foundations of computing. It is important that one understands how to design them, and analyze them for correctness and efficiency. It is important also that one recognizes whether a problem is intractable so one does not naively seek an efficient algorithm when none may exist. The intent of this course is to provide students with fundamental training in these aspects.

| | |
|---|---|
| Target audience | Graduate students in engineering at Waterloo |
| Recommended prior knowledge | ECE 250 or an equivalent<br>`http://www.ucalendar.uwaterloo.ca/1920/COURSE/course-ECE.html#ECE250` |
| Lectures | Online, pre-recorded which you watch at your leisure.<br>Videos will be posted weekly by midnight Mondays.<br>(This means you disregard any timeslot for lectures that shows up in the schedule of classes.) |
| Instructor | Mahesh Tripunitara, `tripunit@uwaterloo.ca`<br>"Mahesh" "Dr. T" "Prof. T"<br>Office hours:<br>TBD |
| TA(s)<br>*For issues regarding assignments only* | TBD |
| Course materials | `learn.uwaterloo.ca` |
| "Textbook" | Tripunitara, "ECE 606 – Algorithms"<br>Available on Learn. May be posted in stages only. |
| Discussions | Self signup as of Sept. 1, 2021 at<br>`piazza.com/uwaterloo.ca/fall2021/ece606` |
| Marking | Weekly assignments 50%<br>Final exam 50% |
| Audit | All deliverables (assignments + final exam) must be met. A mark of 50 on the course must be achieved. |
| If you are waiting to get into the course | There is usually a precipitous drop in enrollment in the first couple of weeks of term. Simply pretend as though you are in the course. Submit assignments via email. Keep checking for open spots to sign up. |
| AccessAbility | `uwaterloo.ca/accessability-services` |
| Academic Integrity | `uwaterloo.ca/academic-integrity`<br>$\longrightarrow$ Students |

# Content Schedule

| *Week of* | | *Topics* |
|---|---|---|
| (1) | Sep 06 | Intro to the course; Discrete math review; intro to Python 3 |
| (2) | Sep 13 | Expressing algorithms; Data structures review |
| (3) | Sep 20 | Properties of algorithms: existence, correctness, efficiency |
| (4) | Sep 27 | Design strategy I: incremental |
| (5) | Oct 04 | Design strategy II: divide-n-conquer |
| (-) | Oct 11 | (nothing; reading week) |
| (6) | Oct 18 | Design strategy III: greedy |
| (7) | Oct 25 | Design strategy IV: dynamic programming |
| (8) | Nov 01 | Randomization, Probabilistic and approximation algorithms |
| (9) | Nov 08 | Non-determinism; computational complexity; the class **NP**; other complexity classes |
| (10) | Nov 15 | Cook- and Karp-reductions; hardness and completeness for a complexity class |
| (11) | Nov 22 | **NP**-complete problems and reductions between them |
| (12) | Nov 29 | Reconciling intractability; common mistakes |
| (-) | Dec 06 | (nothing; already done with 12 weeks) |
| Dec 9 – 23, final exam | | |

# Assignments

There will be weekly assignments, for a total of 12 assignments across the course. Each comprises a few problems that the TAs will mark. There may be problems in the assignment that involve programming in Python 3; each such problem will be annotated with "[**python3**]." Assignments will be published by midnight every Tuesday. Each is due in a week, i.e., by midnight the following Tuesday. Written solutions must be typeset, or written legibly and scanned, and uploaded as PDF. Some subset of the problems on each assignment will be marked by us. This subset will not be announced beforehand.

Lateness policy: no late submissions accepted.

Collaboration policy: you may collaborate with your colleagues when working on your assignments in that you can discuss ideas with one another. However, your final submission must be your own. That is, when you sit down to write your solutions down, you should do so on your own. Any sources you use, whether they are your colleagues, books, papers or online resources, should be appropriately credited in your submission. There is no penalty for utilizing such (re)sources, provided they are credited explicitly. Otherwise, it is regarded as plagiarism, and is an academic offence.

Originality detection: We will be using the following software in this course to check that your submissions are indeed original, and not plagiarized.

> **Turnitin.com**: Text matching software (Turnitin®) may be used to screen assignments in this course. Turnitin® is used to verify that all materials and sources in assignments are documented. Students submissions are stored on a U.S. server, therefore students must be given an alternative (e.g., scaffolded assignment or annotated bibliography), if they are concerned about their privacy and/or security. Students will be given due notice, in the first week of the term and/or at the time assignment details are provided, about arrangements and alternatives for the use of Turnitin® in this course.
>
> It is the responsibility of the student to notify the instructor if they, in the first week of term or at the time assignment details are provided, wish to submit the alternate assignment.

# Final Exam

The final exam will be published similarly as the assignments. However, you will have 24 hours only to turn in your solutions. Your submission is expected to be your own; you are not to collaborate with anyone, in the course nor outside. However, you're free to use any resources, e.g., books/papers/online resources, that you like. You should credit any (re)source you use in your solutions.