ECE 606, F'19, 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.

Prerequisite	ECE 250 or an equivalent `http://www.ucalendar.uwaterloo.ca/1920/COURSE/course-ECE.html#ECE250`
Lectures	[redacted]
Instructor	Mahesh Tripunitara, `tripunit@uwaterloo.ca` "Mahesh" "Dr. T" "Prof. T" Office hours: [redacted] Or by appointment
TAs *For issues regarding assignments only*	Puneet Gill, `p24gill@uwaterloo.ca` Jon Shahen, `jmshahen@uwaterloo.ca`
Course materials	`learn.uwaterloo.ca`
"Textbook"	Tripunitara, "ECE 606 – Algorithms" Available on Learn. May be posted in stages only.
Discussions	Self signup at `piazza.com/uwaterloo.ca/fall2019/ece606`
Marking	Weekly assignments 50% Final exam 50%
AccessAbility	`uwaterloo.ca/accessability-services`
Academic Integrity	`uwaterloo.ca/academic-integrity` $\longrightarrow$ Students

Content Schedule

Lecture		*Topics*
(1)	Sep 04	Intro to the course; Discrete math review; intro to Python
(2)	Sep 11	Data structures review; algorithm correctness; non-existence result
(3)	Sep 18	Algorithm efficiency; Landau symbols
(4)	Sep 25	Subroutine and recursive calls; the incremental design strategy
(5)	Oct 02	The divide-n-conquer strategy
(6)	Oct 09	The greedy strategy
Oct 16, reading week		
(7)	Oct 23	The dynamic programming strategy
(8)	Oct 30	Probabilistic algorithms
(9)	Nov 06	Non-determinism; computational complexity; the class **NP**; some other complexity classes
(10)	Nov 13	Cook- and Karp-reductions; hardness and completeness for a complexity class
(11)	Nov 20	**NP**-complete problems and reductions between them
(12)	Nov 27	Reconciling intractability; common mistakes; review
Dec 6 – 21, 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, and additional problems for you to practice the material, but for which you do not turn in solutions as those will not be marked. 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 on the day of each lecture. They are due by 11:59pm the following Tuesday. Written solutions must be typeset, or written legibly and scanned, and uploaded as PDF.

Lateness policy: no late submissions accepted.

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

Closed book, closed notes, no calculators or other devices allowed. All you will need it a pencil or pen. I may provide you a cheat-sheet that I put together.