COMP 6902: Computational Complexity

Winter 2023

Instructor: Noah Fleming

Email: nfleming@mun.ca

Office: EN-2018

Lectures: TBA

Lecture Room: TBA

Office Hours: TBA (or by email appointment)

Course Description

How difficult is it to solve a given computational task? This is the fundamental question of computer science. On one hand we have designed some remarkable algorithms which are able to handle a wide variety of important problems, which have reshaped many aspects of modern day life. On the other hand, there remain many important problems for which we have yet to discover a fast algorithm. Is this inherently the case — do there not exist fast algorithms for these problems? What about for the problems for which we do have fast algorithms — to what extent can these algorithms be improved? More generally, can we say that one problem is harder to solve than another? In this course we explore how we can answer these questions.

Tentative Marking Scheme

- Midterm 1: 42.5%.
- Midterm 2: 42.5%.
- Assignments (3): 15% total.
- Bug Bounty: There will be a bounty (free marks!) on finding errors in slides, assignments, etc. If you found an error in marking your assignment (e.g., an incorrect solution marked as correct) then you can get additional credit for it.
  Eligible bugs: Are conceptual/theoretical errors, not typos (e.g., a missing non-trivial case in an algorithm or proof, an incorrect step, or an unsolvable or trivial solution to a problem).
  Getting Marks: For bugs in course material, please post the bug on the discussion forum on Brightspace: the first person to find and post the bug will get its bounty. For bugs in marking, please email me directly.

Prerequisites

This course assumes proficiency with the basics of computer science: algorithm analysis and asymptotic notation, pseudocode, discrete math, and basic data structures. See the undergraduate courses COMP 1002 and COMP 2002 which contain most of the prerequisites that you will need.

Tentative Course Outline

- Turing Machines and the Church Turing Thesis.
- Time and space complexity measures and complexity classes (P, NP, EXP, L, NL, etc).
- Hierarchy theorems.
- The polynomial hierarchy.
- Circuit Complexity
- Proof Complexity

Textbooks and Reference Materials

No textbook will be required for this course, however below are some reference materials that may be helpful.

- *Computational Complexity: A Modern Approach* by Arora and Barak.
- *Introduction to the Theory of Computation* by Sipser.

Collaboration and Plagiarism

First of all, the work you submit *must be your own*. You are encouraged to work together on *assignments* and ask questions about them on the discussion forum. However, you must work out all of the answers that you submit on *exams by yourself*. If you come across an answer to a similar problem while researching a question online or in a textbook, you must reference the source and restate the solution in your own words in order to receive full marks.

Plagiarism is a serious academic offence and will be dealt with accordingly. Posting any course content (assignments, exams, practice problems) on the internet, with or without solutions, or using services such as Chegg is a *serious academic misconduct* which will be reported.