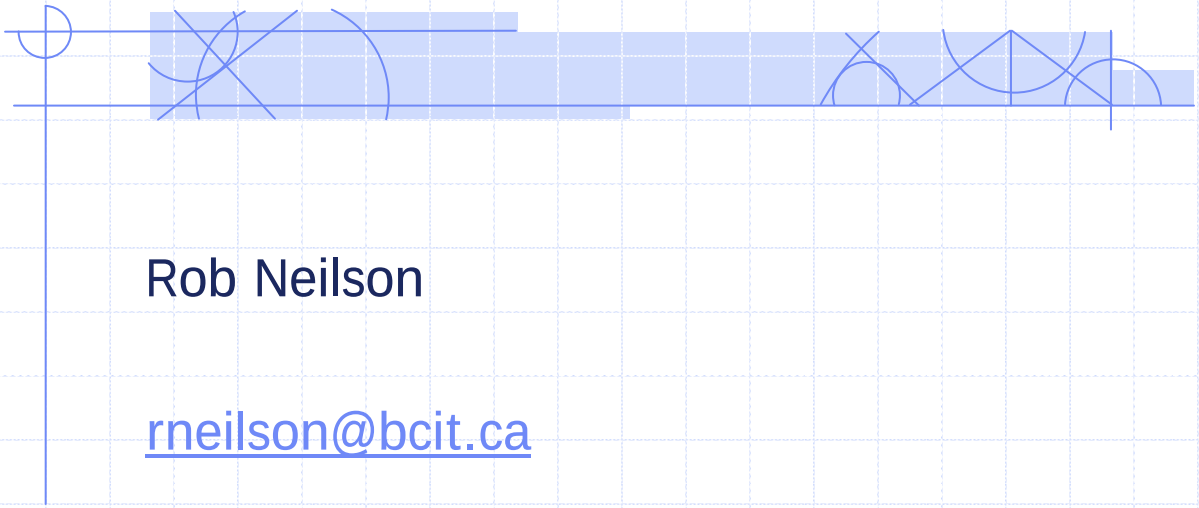
COMP 3760: Algorithm Analysis and Design

Lecture 1: Course Overview



Rob Neilson

rneilson@bcit.ca

Instructor

- Instructor
  - Rob Neilson SW2-125
- Office hours
  - Monday 1:30-2:20
  - Wednesday 1:30-2:20
  - Friday 11:30-12:30
  - *note: these times may change!*

Important Stuff

- Textbook (required)
  - Introduction to The Design & Analysis of Algorithms, 2nd Edition, Levitin, Addison Wesley
- Course web site
  - The course is maintained/managed on WebCT.
  - WebCT will be used for course forums/discussion groups, announcements, posting of notes, and submission of assignments.
- Grading
  - Lab Assignments 20%
  - Major Assignments 20%
  - Midterm 30%
  - Final Exam 30%

Learning Outcomes

Upon successful completion of this course, you should be able to:

- Describe common algorithms
- Analyze pseudo-code using the Big-Oh notation
- Deduce the complexity of a program by running different experiments
- Design new algorithms using several techniques e.g. Divide and Conquer, Greedy, Graph techniques, etc
- Implement newly designed algorithms in Java
- Recognize different types of problems and how to go about solving them
- Argue the correctness of algorithms

Course Prerequisites (1)

- Java (Comp 2526)
  - ***You need to be able to program in Java***
  - java is used in the lab assignments
  - you need to be able to use java in your lab
  - you need to be able to design, write, and debug programs
  - this is not a programming language course
  - you should use good program design techniques as taught in other CST courses
  - use any editor/IDE you want
  - most programs will involve console I/O (no Swing)
  - I am not a Java teacher; you are expected to know Java

Course Prerequisites (2)

- Discrete Math (Comp 2121)
  - a portion of this course is theoretical in nature
  - we use math to determine the relative efficiency of various algorithms
  - you will need to be able to set up and solve simple polynomial equations
  - you will need to be able to set up and solve summation formulas
  - you need to be comfortable working with logarithms (mostly base 2)

Course Prerequisites (3)

- C programming is also good to have
  - many algorithms are described (and often implemented) in straight-line procedural code
  - there are lots of C reference implementations on the internet for common algorithms

Comp 3760 – Course Strategy

- Each week will focus on a specific "core topic“
  - there are way more topics than can be covered in a 15 week course
  - each year I try to change the topics a bit
  - most topics are discussed and supported in the textbook

- Student strategy for success:

(a) read the text and do sample problems (homework)
(b) participate in lessons (lectures)
(c) do the in-class lab assignments
(d) complete the major assignments

Students who allocate time for all the above activities generally do quite well in this course; conversely, most students who do not pass are rarely seen in lecture, skip lab, and do not complete the assignments!!

Reading and Homework

Goal:

The goal of reading and homework is to impart the foundational knowledge required to learn the core topic.

If you do not do the reading and homework, you will probably fail the exams.

1. Readings and homework will be assigned during lecture, about one week in advance. It is also listed in the course schedule on webct.
2. Homework questions are (for the most part) taken from the textbook.
3. Homework questions are required to be completed before the start of your weekly lab (hand in a paper copy which I can grade in class).

   If you do not complete the homework you will have to work on it while others start the lab assignment. Answers will be posted on webct.
4. A portion of your lab grade is based on your homework being done and somewhat correct.

Lessons

Goal:

The goal of the lessons is to re-enforce the foundational knowledge, to demonstrate/explain how this knowledge can be applied/used, and to impart an appreciation of why the topic is relevant/important.

1. lessons assume that reading/homework was done before class
2. lessons will include time for ad-hoc Q&A on the readings
3. I try not to simply regurgitate info from the textbook
4. in each lesson I try to involve the class (you) in
   - solving a problem, or
   - reviewing an implementation, or
   - investigating the application of a technique
   - studying an algorithm
5. each week will end with a sample exam question; solutions to these should be posted and discussed by you all in the online course forum
6. all class notes will be posted on WebCT ; hand-written stuff from lectures will be scanned and posted

Labs

Goal:

The goal of the labs is to teach you to apply the new techniques/knowledge in a controlled environment.

1. labs assume that reading/homework was done before class, and lessons were attended
2. at the start of some of the labs, students will be selected to present/discuss their answers to the homework questions
3. the actual lab exercise will not be distributed until you hand in your homework
4. labs will be done individually (unless otherwise specified)
5. labs will involve solving small problems (hand-written solutions or writing small computer programs)
6. lab assignments are due before the end of class, and will be graded by the attending lab instructor; there are no extensions for lab assignments

Major Assignments

Goal:

The goal of the assignments is to use the new techniques to solve a larger more realistic problem.

1. assignments build on techniques as they are introduced throughout the course, with earlier assignments being easier than later ones
2. assignments involve integration of the new techniques/topics/knowledge with skills/knowledge acquired in previous courses
3. not all skills/knowledge required for an assignment will be covered in class
   - some material may be prerequisite knowledge
   - sometimes the solution will require research and/or ingenuity on the part of the student
4. assignments will be done in pairs (or individually if you prefer)
5. the grading of the assignments will focus on your ability to apply the techniques learned in class, and on the correctness of your solution
6. an overview of grading criteria will be published with the assignment
7. there will be 3 major assignments during the course; the exact details have not been decided yet

Examinations

Goal:

The goal of the examinations is to assess the students (a) foundational knowledge (b) ability to apply the skills/techniques

1. exams are closed-book, with 1 page of notes being allowed (so you don't have to memorize formulas etc)
2. sample exams will be provided so that you know what to expect
3. midterm is 2 hours, and covers the first half of the course
4. final is 2 hours, and covers the second half of the course

Schedule of Topics

Week #	*Week Starting*	*Topic*
1	Sept 3	Introduction
2	Sept 8	Efficiency Analysis
3	Sept 15	Brute Force
4	Sept 22	Hard Problems
5	Sept 29	Input Enhancement
6	Oct 6	Fundamental Data Structures
7	Oct 14	Heaps and Priority Queues
8	Oct 20	Midterm Exam
9	Oct 27	Graphs
10	Nov 3	Graph Traversal
11	Nov 10	Topological Sort
12	Nov 17	Spanning Trees
13	Nov 24	Transitive Closure
14	Dec 1	Shortest Path
15	Dec 8	Final Exam

Note: this schedule of topics is subject to change

Homework and Reading

(due start of lab next week)

Reading:

- chapters 1.1, 1.2, 1.3, 1.4
- chapters 2.1, 2.2, 2.3

Homework:

- chapter 1.1, page 8, question 8
- chapter 1.2, page 18, question 9
- chapter 1.3, page 23, question 1
- chapter 1.4, page 38, question 1
- chapter 2.1, page 52, question 2
- chapter 2.3, page 67, question 4

The End