CS-251A Data Structures and Algorithms

Hints on How to Pass this Course

1. **Welcome!**

   Welcome to Computer Science 308-251A: Data Structures and Algorithms. This course is an introduction to the design and analysis of algorithms. Below is a description of the main ingredients involved in taking this course with hints for passing, enjoying and making the best out of the course.

2. **No Programming:** This course is NOT a programming course and there will not be any programming assignments. The course is instead concerned with the principles behind the design of efficient algorithms for solving a variety of different classes of problems. It is concerned with high-level techniques for designing algorithms in different contexts. It is NOT concerned with the details of implementation on real computers with real programming languages. We will use idealized computers (mathematical models of computers if you like) such as the Real RAM (Random Access Machine with real numbers as inputs) to measure the computational complexities of our algorithms.

3. **Algorithms for Humans:** Although in this course we are concerned with designing algorithms that are ultimately intended for machines to process, we are NOT interested in communicating with the machines but rather with each other. We are interested in communicating with humans about algorithms. Therefore we will NOT use **code** in this course. Code was designed to be understood my machines not humans. We will describe algorithms in a natural language such as English and in the language of mathematics as has been done for thousands of years. Furthermore mathematics should not be confused with mathematical notation. The essence of mathematics lies in the precision of concepts and not symbolic notation. Mathematical notation will be minimized. Algorithms described in code will get **ZERO** marks. Students using mathematical notation gratuitously will be penalized. Mathematical notation should be used only when absolutely necessary if it helps the reader. Remember that storage space is no longer a problem and we need not save precious space at the cost of making things difficult to read. In this course we will learn how to use English in a precise way.

4. **Correctness of Algorithms:** One of the most important properties of an algorithm is its correctness. In fact this is so important that many people refuse to call a procedure an **algorithm** if it is not guaranteed to give the correct solution. Many “algorithms” inhabiting the computers in society at large are incorrect. This can cause great cost and tragic suffering. For example, patients have been killed by overdoses of radiation because of incorrect algorithms. The American astronauts aboard the space shuttle Challenger were killed because of an incorrect geometric algorithm used in rebuilding the booster rockets. In fact a nuclear world war could be caused by an incorrect algorithm. Therefore it is very important to design correct algorithms. In later computer science courses students often have trouble proving the correctness of algorithms. Therefore this course will concentrate on various techniques for proving the correctness of algorithms.

5. **Induction Proofs:** In later courses students have the most trouble with **induction** proofs. In a very important sense this is the most relevant method of proof for computer scientists since it

lies at the heart of recursion and divide and conquer, two fundamental techniques for obtaining efficient algorithms. Therefore this course has as one goal for the students to master proofs by induction. Induction proofs will be a recurring theme in the course.

6. **Course Web Page:** Students should consult the course web page regularly. It can be found at: http://www-cgrl.cs.mcgill.ca/~godfried/teaching.html. This page contains several sections we will now describe.

   a) **Logistics Information:**

   This section of the course web page details the performance evaluation of the students, the coordinates of the lecturer and teaching assistants, the text book used in the course and similar matters.

   b) **Calendar and Bulletin Board:**

   This web page contains the calendar of events for the course such as the in-class exam and quizzes, the dates the assignments are handed out and due, and a very short title of each lecture. The **bulletin board** at the beginning contains messages from me to you of important day to day events, matters related to the assignments, etc and must be checked every weekday.

   c) **Lecture Descriptions, Homework and Play:**

   This web page contains a detailed list of topics covered in every lecture. It contains the **reading assignments** made every week from the text book, handouts and web material. It also contains **suggested play**. While this section is not required, students are strongly encouraged to play with the applets suggested in this section. Students who do will be at an advantage as it greatly facilitates learning the material in this course.

   d) **Course Outline:**

   This page contains an skeletal outline of the course contents.

   e) **WEB Course Contents**

   This page contains much of the material that we will cover in the course that is not contained in the course text book (Udi Manber: Introduction to Algorithms). Many of the reading assignments will concern pages in this section. Here you will also find alternate descriptions of material that is in the text book. It sometimes helps students to learn when they read different explanations of the same thing by different people. This web page also contains the applets for the suggested play section of the assignments.

7. **Class Quizzes and Exam:**

   There are four in-class quizzes and one in-class exam during the course. All these are **closed book**. However, I am not interested in measuring your memory. Computers are much better than we humans at that. The tests will measure your ability to think creatively. Students often ask: *what material will be covered in the test?* My answer is always the same. I don't want to insult your intelligence by asking you to regurgitate material we have covered. A machine can do that much better than we can and I don't want to treat you like machines. I want to test you on what machines cannot do (yet), and that is to generalize or apply your knowledge to solve new problems in new contexts. Therefore my tests will always cover everything we have **not** covered in the course. The material I learned as a student in university concerning radios, cathode ray tubes and transistors is useless to me now. Many of the things I teach now did not exist ten years ago. However, the creative problem solving techniques and critical thinking skills I learned as a student can be applied to things other than transistors, such as the latest trends in computational geometry. How can you do well on my tests? University is (still, thank heaven) a place for discussing knowledge in a free and open atmosphere and I encourage you to discuss all matters of the course with your class mates. On the other hand the modern harsh reality dictates that university is also a certification method for certain skills - hence the tests. The best way to study for the tests in my course is to do the assignments and exercises **on your own.** Only personal discovery leads to true understanding. The text book by Udi Manber also contains many exercises at the end of each chapter **with** answers at the back of the book. Doing these exercises is an excellent way to learn how to design algorithms.

8. **Assignments and Late Assignment Policy:**

   Please write legibly. If the TA cannot read your assignment you will get **ZERO**. Whenever you are describing an algorithm you **MUST** always include a proof of correctness. **NEVER** say that something is obvious or “clearly....”. Obviousness is always the enemy of correctness and sometimes insults the reader as well. Always give clear simple short arguments. Long convoluted sentences increase the probability that they are wrong. Pretend you are explaining things to your ten-year old sibling. Don’t guess. If you are not sure of something say so. Clear thinking with a partial answer will be rewarded. Cloudy thinking with a full answer will be penalized.

   I am unforgivingly strict with late assignments. Assignments **MUST** be handed in on the day due and **BEFORE** the lecture begins. Every day that the assignment is late, starting with the day due and including weekend days, will result in a penalty of 10% of the original worth of the assignment.

9. **Teaching Assistants:**

   The TA’s are responsible for marking the assignments and the quizzes and exam. If you have any questions about how you were marked please go see them. If you have general questions about the course material please see them too. They are there to help you.

10. **Good Luck!**

    If for some reason the TA’s cannot help you please come and see me. In any case please come by sometime to say hello and tell me how you are doing in the course. My door is wide open during office hours and I would love to meet you.