Computer Science 2300: Introduction to Algorithms

Spring 2012

RPI, Spring 2012
Instructor: Sanmay Das
Office: Lally 302
Phone: x2782
Office hours: Mondays from 3:30-5:00 PM, and by appointment.

1 Course Description

1.1 Staff

The TAs for the class this semester are Bhavana Malhotra and Fredrick Sisenda.

	Bhavana Malhotra	Fred Sisenda
e-mail	malhob@rpi.edu	sisenf@rpi.edu
Labs	W 10-12 and 12-2	W 2-4 and 4-6

There will also be several undergraduate programming mentors who will help out with labs.

1.2 Overview

This is a foundational course in the design and analysis of algorithms and the data structures they are coupled with. While we will concentrate on the theoretical design and analysis of algorithms and data structures, we will reinforce the theory with examples, laboratories, and applied homework questions.

1.3 Learning Outcomes

The goal of this course is to provide a strong foundation in algorithms and data structures in preparation for jobs in industry or for more advanced courses. Algorithms are the basic language of computer science. After taking this course, you, the student, should be able to:

- Prove the correctness of, and analyze the running times of, different algorithms.
- Use different algorithm-design techniques, including, but not limited to, greedy, divide-and-conquer, and dynamic programming techniques, to solve particular problems.
- Model real problems abstractly using the language of graphs and flows.

- Solve problems by reducing to other problems whose solution is known, and show that problems are hard by reducing from other problems.
- Make intelligent decisions about alternative data structures and algorithmic techniques in the context of practical software problems, choosing from existing data structures and algorithms or designing your own when necessary.

1.4 Prerequisites

The official prerequisites for this course are CSCI 1200 (Data Structures), and Math 2800 (Discrete Structures).

You should be familiar with basic data structures and algorithms from CS II, including arrays, linked lists, trees, iteration, and recursion. The analysis of algorithms also relies heavily on discrete mathematics, as covered in Discrete Structures. In particular, you should be familiar with mathematical induction, basic probability, and permutations and combinations.

A working knowledge of C++ is required for completion of the labs and some aspects of the problem sets. Note, however, that this is not a course about C++; the course focuses on the fundamental issues in the mathematical analysis of algorithms. Nothing about the C++ language will be discussed during the lectures – it is up to you to ensure that you are sufficiently comfortable with the language.

2 Policies

2.1 Website and Communication

We will make extensive use of electronic communication and the course website. You are responsible for checking the course website regularly for announcements and course materials, as well as your e-mail for communications related to the class. The course website is located at:
`http://www.cs.rpi.edu/~sanmay/cs2300/`

2.2 Lectures

Lectures will be approximately 90 minutes long. They will be held from 12-1:30 on Mondays and Thursdays in Amos Eaton 214. You are responsible for all material covered and announcements made in lecture.

2.3 Labs

The purpose of labs is to help you with the course material in a smaller setting. You will be required to complete a small assignment for each lab. This assignment will be handed out in advance, and ideally, you will finish the assignment in advance of the lab. However, you can get help from the TAs during lab if you are unable to finish in advance.

See the class hour schedule for the time and place of your lab section. Labs will be held weekly and attendance to scheduled labs is required. Those who show up late for lab or those who leave without completing the lab risk not receiving full credit. Labs cannot be made up. The exams will contain material covered in labs. Due to the size of the sections, you **must** attend your assigned lab section. The TA will take attendance and you will not receive credit unless you attend your assigned section.

Frequently, labs will be replaced with combined recitation/office hours held by the TAs. These sessions are intended to provide you with support for honing your problem solving skills, as well as helping out with homeworks.

2.4 Homework

The homeworks will typically consist of more difficult problems than the labs. They are one of the most important parts of the class, and they carry substantial weight in your final grade. This is re-iterated in the policy on academic integrity below, but we cannot stress it enough: *any use of material from previous versions of this class, or from solution manuals from the textbooks will lead to disciplinary action* (trust us, it's not that hard to figure out when this happens).

Homeworks will typically be due **at the beginning** of class. Electronic submissions will **not** be accepted. Each student will be given a budget of five *late days* that they can use to turn in homeworks late. A late-day can be used without explanation to extend a homework submission deadline by 24 hours, but no more than two late-days can be used on any one homework. You must notify your TA when you use late days and set up an alternative way to submit your homework.

Any part of a late day that you use counts as a full late day. For example, if you do not submit your homework until the end of lecture instead of the beginning, that counts as a full late day. If you submit 26 hours late, you will have used two late days. You are responsible for keeping track of your usage of late days. **Use your late days wisely, if at all.** This late-day policy is intended to cover unanticipated things like minor sickness, exams in other classes, etc. so that you do not have to ask for extensions. Once you have used up your budget of late days you will not be allowed to turn in homeworks late for any reason without a written note from the office of the Dean of Students. **Late days are not applicable to labs!**

2.5 Exams

There will be two midterm exams during the semester and a comprehensive final exam during finals period. We will not provide makeups unless the absence is excused by the Dean of Students.

2.6 Assessment and Course Grade

Your overall course score will be determined using the following weights:

1. Labs: 20%
2. Homework: 30%
3. Midterm 1: 10%
4. Midterm 2: 15%
5. Final exam: 25%

Students will be graded on a curve. There are no fixed grade thresholds. We will give an approximate grade breakdown in the middle of the semester, but **you are responsible for keeping track of your own grades on WebCT!**
If you would like to appeal your grade on any of the assignments or exams, you may do so within **one week** of the assignment or exam being handed back. In order to appeal the grade, please

provide a detailed written statement explaining why you believe the assigned grade is incorrect, in addition to the assignment or exam. Your **entire** assignment or exam will be regraded and your grade may go up or down, or it may stay the same.

2.7 Collaboration and Academic Integrity

The statement below is based partially on a model statement from the provost.

Student-teacher relationships are based on trust. For example, students must trust that teachers have made appropriate decisions about the structure and content of the courses they teach, and teachers must trust that the assignments that students turn in are their own. Act which violate this trust undermine the educational process. The Rensselaer Handbook of Student Rights and Responsibilities *defines various forms of Academic Dishonesty and you should make yourself familiar with these.*

In this class, you are allowed to collaborate on assignments to the following extent. You are welcome to discuss problems (both written and programming) with each other and to take your own notes during these discussions. However, you must write up solutions on your own. You must write, on the assignment, the names of students you discussed each problem with, and any external sources you used in a significant manner in solving the problem. Lack of citation of a source is a serious violation of this policy. In programming assignments, please do not look at the code of any other student in the class, or allow any other student to look at your code. In addition, any use of material from previous versions of this class, or from solution manuals for the textbooks, is cheating.

Submitting an assignment that is in violation of this policy will automatically lead to receiving no credit for the assignment and a reduction of at least one grade modifier (e.g. from B to B-, or B- to C+) beyond that in the overall course grade. However, depending on the circumstances, it could also lead to harsher penalties, for example, a failing grade in the class and initiation of the Institute-wide disciplinary process.

3 Textbooks and Syllabus

3.1 Textbooks

There are two required textbooks:

1. *Algorithms* by S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani, McGraw-Hill.
2. *Introduction to Algorithms* by Cormen, Leiserson, Rivest, and Stein, MIT Press / McGraw-Hill.

3.2 Syllabus

This syllabus is a broad outline and may change based on time constraints or other factors.

1. Basics of algorithm analysis
2. Divide-and-conquer algorithms: multiplication, mergesort, quicksort, median-finding, matrix multiplication
3. Basic graph algorithms: depth-first and breadth-first search, connected components, Dijkstra's algorithm.
4. Heaps and heapsort

5. Minimum spanning trees and the disjoint set data structure
6. Trees
7. Hashing
8. Dynamic programming
9. NP-complete problems