Computer Science 2300: Data Structures and Algorithms

Spring 2009

RPI, Spring 2009
Instructor: Sanmay Das
Office: Lally 302
Phone: x2782
Office hours: Mondays from 1:30-2:30 PM, or by appointment.

1 Course Description

1.1 Staff

The TAs for the class this semester are Eyuphan Bulut and Ashok Sukumaran.

	Eyuphan Bulut	Ashok Sukumaran
e-mail	bulute@cs.rpi.edu	sukuma@rpi.edu
Labs	W 2-3:50 and 4-5:50	W 10-11:50 and 12-1:50

There will also be several undergraduate TAs 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 Objectives

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 should be able to prove the correctness of and analyze the running times of different algorithms. You should also be able to use different algorithm-design techniques to solve particular problems.

Data structures and algorithms also form a major component of any software system. After taking this course you should be able to make intelligent decisions about alternative techniques in the context of practical 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 (Computer Science II), 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.

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 or in the textbooks – 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 (1) announcements made in lecture, (2) announcements and materials posted on the course website, and (3) class-related communications sent to your e-mail account. The course website is located at: `http://www.cs.rpi.edu/~sanmay/cs2300/`

2.2 Lectures

Lectures will be held from 12-1:30 on Mondays and Thursdays in Darrin 330. 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 may 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.

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. 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 the TA responsible for the homework 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 one midterm exam 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: 35%
3. Midterm exam: 20%
4. Final exam: 25%

In the worst case (from the student's perspective) grades will be assigned as follows:

- 90-100%: A range (A, A-)
- 80-90%: B range (B+, B, B-)
- 70-80%: C range (C+, C, C-)
- 60-70%: D range (D+, D)
- Below 60%: F

We may *lower* the thresholds described here depending on the performance of the class. We will not raise the 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 it being handed back. In order to appeal, 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 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.

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 (2nd Ed)* by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, 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