Course Syllabus
CS375: Design & Analysis of Algorithms
Spring 2017

Course Time and Location

Lecture hours: **A0**: 1:10pm-2:10pm (MWF); **B1**: 2:20pm-3:20pm (MWF)
Lecture location: UU 103
Lab hours: **A50**: 2:50pm-4:15pm (T); **B51**: 4:25pm-5:50pm (T)
Lab location: **A50**, SW 214; **B51**: SW 212

Instructor and Teaching Assistants

Instructor: Lei Yu
Office: Q5, Engineering Building
Office hours: 3:20pm – 4:00pm, MWF, or by appointment
Email: lyu@cs.binghamton.edu
Webpage: http://www.cs.binghamton.edu/~lyu/index.html

Teaching Assistant: Andrew Cohen (primary for labs)
Office: N21, Engineering Building
Office hours: 10:00am - 11:00am, MF, or by appointment
Email: acohen13@binghamton.edu

Teaching Assistant: Ding Ding (primary for A0 session)
Office: N21, Engineering Building
Office hours: 12:00pm - 1:00pm, TR, or by appointment
Email: dding1@binghamton.edu

Teaching Assistant: Dinuni Fernando (primary for B1 session)
Office: N21, Engineering Building
Office hours: 11:00am – 12:00pm, TW, or by appointment
Email: dferna15@binghamton.edu

Course Description

Analysis of common algorithms for processing strings, trees, graphs and networks. Comparison of sorting and searching algorithms. Algorithm design strategies: divide and conquer, dynamic programming, greedy, back tracking, branch and bound. Introduction to NP-completeness. Required activity includes student presentations.

Prerequisites

CS 240: Data Structures and Algorithms
Math 314: Discrete Mathematics

CS375 assumes students have mastered:

1. A high level language such as Java, C or C++.
2. Basic data structures such as arrays, linked lists, trees, heaps, and graphs.
3. Basic mathematical concepts such as:
    a. Logarithms and exponents
    b. Arithmetic and geometric series
    c. Combinations and permutations

d. Limits and derivatives
e. Proof techniques such as induction, direct proof, proof by contradiction, etc.

Course Objectives

This course is designed to provide a solid foundation and background in the analysis and design of computer algorithms. In particular, upon successful completion of this course, you will be able to:

- use critical thinking for problem solving
- implement algorithms efficiently and correctly
- argue algorithm correctness
- analyze time complexity of algorithms
- know and use common algorithms
- learn to design efficient algorithms using well-known methods
- describe effectively, in writing and in an oral presentation, an algorithm and its implementation

Textbook

- T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to Algorithms*. 3rd edition, The MIT Press, 2009.

It is highly recommended that you purchase your own copy of this textbook and keep it even after this semester. This book will serve as one of the single most useful and important reference books beyond this course.

Reference Books

- R. Neapolitan and K. Naimipour. *Foundations of Algorithms*. 4th edition, Jones and Bartlett Publishers, 2011.
- Kleinberg and Tardos. *Algorithm Design*. Addison Wesley, 2005.
- A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *Data Structures and Algorithms*. Addison-Wesley, 1984.
- S. Baase. *Computer Algorithms- Introduction to Design and Analysis*. 3rd edition, Addison-Wesley, 2000.
- G. Brassard and P. Bratley. *Fundamentals of Algorithmics*. Prentice Hall, 1996.
- E. Horowitz, S. Sahni and B. Rajasekaran. *Computer Algorithms C++*. Computer Science Press, 1996.
- R. Neapolitan and K. Naimipour. *Foundations of Algorithms Using C++ Pseudo Code*. 3rd edition, Jones and Bartlett Publishers, 2004.
- R. Neapolitan and K. Naimipour. *Foundations of Algorithms using Java Pseudo Code*. 3rd edition, Jones and Bartlett Publishers, 2004.
- R. Sedgewick. *Algorithms in Java, Part 1-4*. 3rd edition Addison Wesley, 2003.
- R. Sedgewick. *Algorithms in Java, Part 5*. 3rd edition Addison Wesley, 2004.
- R. Sedgewick. *Algorithms in C++, Part 1-4*. 3rd edition Addison Wesley, 2002.
- R. Sedgewick. *Algorithms in C++, Part 5*. 3rd edition Addison Wesley, 2004.
- R. Sedgewick and P. Flajolet. *An Introduction to the Analysis of Algorithms*. Addison Wesley, 1996.
- S. S. Skiena. *The Algorithm Design Manual*. Springer-Verlag 1998.

Main Topics

No.	Topic	Cormen Textbook Chapters
1	Time complexity, insertion sort and	Chapters 1, 2, 3

	merge sort, asymptotic growth functions	
2	Solving recurrences, divide and conquer algorithms	Chapter 4
3	Heaps, heapsort, binary search trees	Chapters 6, 12
4	Sorts	Chapters 7, 8
5	Backtracking	
6	Dynamic programming	Chapter 15
7	Greedy method	Chapter 16
8	Amortized analysis	Chapter 17
9	Graphs, minimum spanning tree, shortest path	Chapters 22, 23, 24
10	NP-Completeness	Chapter 34

Lecture Notes

Slides for each lecture will be posted on myCourses before lectures. *Lecture slides do not substitute for class attendance*, since (i) they will not be complete and (ii) significant parts of lectures, including discussions and in-class exercises, may not come from the lecture slides.

Grading

1. You grade will be based on

Three exams	40% (Exam I 10%, Exam II 10%, Final 20%)
7-8 short quizzes	10% (2% each, after dropping 2-3 lowest grades)
Four theoretical assignments	20% (5% each)
Two programming assignments	20% (10% each)
Project and presentation	10%

2. Exams

Exams will be in class, closed notes, and closed book, unless otherwise specified (unlikely). Exam I will take place during (or very close to) Week 6, Exam II during (or very close to) Week 10, and the Final Exam during the Final Exam Week. Exams II does not explicitly test the materials covered in Exam I, but we expect that you can recall and utilize the most fundamental and important knowledge for topics that were covered in Exam I. The Final Exam will be a comprehensive exam, with a primary focus on the materials that were introduced during the last phase of the semester.

Students MUST take each exam on the scheduled date and time. There will NOT be makeup exams.

3. Quizzes

Some of the quizzes are pop-up quizzes, while others are pre-announced with a scheduled date in class or lab. If you arrive late to a quiz you will not be able to take it. You may *not* "make up" a quiz by attending the other section. If you regularly miss classes or labs, your overall quiz score will be negatively affected. We

will only consider the top 5 of the quiz scores you received in deciding your quiz total score.

4. Theoretical assignments

- Much of what one learns in this course comes from solving problems.
- Start working on the homework early.
- Most questions require both knowledge of the material and problem solving ability.
- If you don't know how to solve a problem, don't give up.
- Make sure that you understand the questions.
- See if you can solve a problem for some simple cases, and then try to find a general solution.
- Try again on the next day.

All solutions of theoretical assignments must be typed (no handwritten solutions) and submitted in hard copy. Advance electronic submission to the TAs is acceptable if the student is expected to miss the class on the due date.

5. Programming assignments

Programs must be written in Java, C++ (or C). Make sure a program compiles on harvey.cc.binghamton.edu and runs correctly.

Please make every effort to program and debug your code on your own. Use gdb, valgrind, IDE debuggers, and other tools. Use Google to learn about compiler and runtime error messages. Consult stackoverflow.com and other useful programming websites. Please begin your assignments early and expect to encounter challenges. Please note that the TA or instructor will not review or debug your code before submission.

You will get a grade of at most 10% if your code does not compile. If the program compiles but it has runtime errors or bugs, your grade will be based on the severity of the bugs.

Please also make an effort to make your programs easy to understand and grade. Grading all assignments in this course is very time consuming! Expect to lose points if you submit a badly documented or commented program. All programming assignments should have:

1. For the program (in a README file):

- Step-by-step instructions on how to compile and run the program.
- The data structures used by the program and how they are implemented.
- Some analysis or discussion of its computation time.
- The classes used and their interaction.

2. For each class in your program

- An explanation of the purpose of the class, and the methods it includes.

3. For each method (function)

- A description of the purpose of each function and an explanation of how it works.
- A description of the purpose, and the assumptions made about each parameter of a function.
- A comment for every variable declaration.

6. Grading disputes and missing grades

In this course, we commonly give partial credit to partially correct answers. Should you dispute a partial credit, please be aware that we will not re-grade a single question in a homework, quiz or exam. ALL partial credits of the work will be re-examined. The new grade may be higher, lower, or stay the same. This new grade will not be changed.

Your grades will be posted on Blackboard. Please check your status on blackboard periodically and make sure that there are no missing grades or errors. A missing grade at the end of the semester will indicate that the work has NOT been done.

Reading Assignments and Review Questions

You will be given a reading assignment for each unit. It may also include material that will not be covered in class but will be included in homework assignments. For some units you will also be provided with some "review questions".

Reading the text in this course is time consuming. To understand the material you may need to read the material more than once. After you read the material try to answer the review questions.

Late Submission Policy

Each assignment is due at the beginning of class on the due date. Any assignment received within the next 24 hours will be penalized by 20% of the full credit; any assignment received within the time between 24 hours and 48 hours pass the deadline is penalized by 40% of the full credit; No assignment will be accepted after 48 hours pass the deadline. Rare exceptions of this policy may be made at the discretion of the instructor under demonstrably circumstances.

Academic Honesty Expectations

Please review the academic honesty document and make sure that you understand it! The link is: http://www2.binghamton.edu/watson/advising/pdfs/honesty-policy.pdf Cheating and copying will NOT be tolerated.

- Each programming or theoretical assignment should start with the following statement:

"I have done this assignment completely on my own. I have not copied it, nor have I given my solution to anyone else. I understand that if I am involved in plagiarism or cheating I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of **0** for the involved assignment for my first offense and that I will receive a grade of **"F" for the course** for any additional offense."

- Each exam will have a first page with the following statement:

"I understand that if I am caught copying or talking during the exam I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of 0 for the involved exam."

Your homework assignment or exam will not be graded unless the statement above is followed by your name.

Collaboration

Students are encouraged to help one another and to form study groups. In Computer Science, you can learn more from your peers than from your instructors and teaching assistants. As long as the help is appropriate, please be generous with your time and expertise when helping fellow students. Doing so is good for you and good for them. You are free to discuss assignments *in general terms* with one another. However, please do not show your work directly to other students. Each student must complete your assignments *individually* (unless indicated otherwise by the instructor). Each of you must write your own code, and you must write up all solutions individually. Students submitting solutions (including code) that are determined to be "too similar" are likely to be punished equally and harshly. We can tell whether you have done the work on your own, so please do the work on your own.

Class and Labs Attendance

Attendance is required. Some of the quizzes will be announced in advance, but some others are not. If you arrive late to a quiz you will not be able to take it. You may *not* "make up" a class or a quiz by attending the other section. If you regularly miss classes or labs, your overall quiz score will be negatively affected.

Computers and Other Electronic Devices

You are allowed to use your laptop/notebook/tablet computers for course related activities (e.g., reviewing lecture materials, taking notes, etc.) except during an exam or quiz. Cell phones must be turned off or in vibrate alert mode during class.