Course Syllabus
CS375: Design & Analysis of Algorithms
Spring 2015

Instructor and Teaching Assistant

Instructor: Weiyi Meng
Office: Q08, Engineering Building
Lecture hours: 2:20pm – 3:20pm (MWF)
Lecture classroom: N 25, Engineering Building
Office hours: 4:00pm – 5:00pm on Monday & Wednesday, or by appointment
Email: meng@binghamton.edu
Webpage: http://www.cs.binghamton.edu/~meng/meng.html

Teaching assistant: Guangyu (Gary) Liu
Office: G25, Engineering Building
Office hours: 11am – noon, Friday, or by appointment
Lab section: LNG 209, 2:50pm – 4:15pm, Thursday
Email: gliu10@binghamton.edu

Coordination with Section A0

The CS Department is offering two sections of CS 375 this semester. Prof. Michael Lewis is teaching Section A0. Prof. Lewis and I plan to coordinate closely, on assignments, rubrics, grading criteria, course materials, exam dates, and assignment due dates. The TA for Section A0 is Madina Zabran (mlatypo1@binghamton.edu). The TAs of the two sections will coordinate the grading of all assignments.

Please send your questions regarding all assignments to cs375-help@cs.binghamton.edu. This email can reach all instructors and TAs of the two sections.

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, 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.

I strongly recommend 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 merge sort, asymptotic growth functions	Chapters 1, 2, 3
2	Solving recurrences, divide and conquer	Chapters 4

	algorithms	
3	Heaps, heapsort, binary search trees	Chapters 6, 12
4	Sorts	Chapters 7, 8
5	Hash Tables	Chapter 11
6	Backtracking	
7	Dynamic Programming	Chapter 15
8	Greedy method	Chapter 16
9	Amortized Analysis	Chapter 17
10	Graphs, minimum spanning tree, shortest path	Chapters 22, 23, 24
11	NP-Completeness	Chapter 34
	Other advanced topics if time permits	

Lecture Notes

Lecture Notes for each chapter, in Powerpoint format, will be posted on blackboard before lectures. I recommend that you print lecture notes beforehand and bring them to class so you can notes easily. *Lecture notes 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 class notes.

Grading

1. Your grade will be based on

- Three exams 39% (13% each)
- 6 short quizzes 15% (3% each after dropping the lowest grade)
- Four programming Assignments 18% (4.5% each)
- Four theoretical assignments 18% (4.5% each)
- Project and presentation 10%

2. Exams. Exams will be in class, closed notes, and closed book, unless otherwise specified (unlikely). Exam 1 will take place during (or very close to) Week 5, Exam 2 during or close to Week 10, and Exam 3 during the Final Exam Week. Exam 3 will primarily cover the materials that were not covered in the first two exams. However, for Exams 2 and 3, even if we do not explicitly test the materials covered by earlier exam(s), we expect that you can recall and utilize the most fundamental and important information for questions that deal primarily with the topics that we intend to cover explicitly on that exam.

You must take each Exam during the one time that we give it; we will not give makeup Exams.

3. Quizzes. The quizzes will usually take place during lab sessions, and we will announce them at least one class period in advance, in class, through Blackboard, or both. Quizzes will be closed notes and closed

book, unless otherwise specified. We will drop your lowest quiz score from your grade. A discussion will be held after each quiz.

You must take each Quiz during the one time that we give it; we will not give makeup quizzes.

4. Assignments

- Much of what one learns in this course comes from solving problems.
- Start working on each assignment 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.

Be clear and neat in your write-ups. Readability of the programs and solutions is as necessary as correctness. Expect to lose points if you provide a badly written and unclear "correct" solution.

Late assignments may sometimes be accepted with penalty, which will typically be 10% per day late. We will not accept assignments more than 2 days after the due date.

Some assignments may contain extra credit questions that may add up to 10% to your overall grade (enabling an overall grade above 100%). These questions are usually more difficult to solve. Students striving for an A are encouraged to work on these problems.

5. Programs and the project. Please make an effort to make your programs easy to understand and grade. Grading all assignments in this course is very time consuming! All programming assignments should have:

1. For the program:
   - A general explanation of the design, and why it is correct.
   - 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.
   - The code should also be well commented.
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.

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

6. Program grading

- 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 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. As a last resort, please email questions to the cs375-help@cs.binghamton.edu or visit office hours. We may ask for output from debuggers and we may or may not decide to help push you in the right direction toward solving your

programming problem. We will do whatever we think will help you become an excellent independent programmer, not what we think will help you get the program done most quickly. Please begin your assignments early and expect to encounter challenges.

- Late assignments may sometimes be accepted with penalty, which will typically be 10% per day late. We will not accept assignments more than 2 days after the due date.

7. 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, primarily from the Cormen textbook. The reading may include material that is *not* covered in class but could be required for assignments. For some units we may also provide some "review questions."

You may need to read the textbook more than once. It is recommended that you familiarize yourself with the material by reading it once before it is covered in class. You will then have some context for understanding lectures as much as possible. After the relevant lectures, you could read the book with more attention to detail, filling in gaps in your understanding. Bring remaining questions to class. You may also decide to read the material again before each exam or quiz.

Academic Honesty Expectations

Please review the academic honesty document and make sure that you understand it! The link is at: http://www.binghamton.edu/watson/about/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 and quiz will have a first page with the following statement:

  "I understand that if I am caught copying or talking during the exam/quiz 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/quiz."

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

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 and attendance will be checked regularly. If you are not present when attendance was checked it will be counted as missing the class. If you arrive late to a quiz you will not be able to take it. You may miss a total of three classes or labs without a verifiable valid excuse. After that your final grade will be reduced 5% for each missing class. If you miss six or more additional (beyond three) classes you automatically fail the class. Please inform the instructor ahead of time by email for any expected or excused absence. You may *not* "make up" a class by attending the other section; we will take attendance only for our own section.

Computers and Other Electronic Devices

You are not allowed to use your laptop/notebook/tablet computers during class unless explicitly permitted. Cell phones must be turned off or in vibrate alert mode during class.