Syllabus: CS575 Design and Analysis of Algorithms

Spring 2016
Department of Computer Science
T.J. Watson School of Engineering
State University of New York at Binghamton

Instructor

- Name: Kyoung-Don Kang
- Office: EB Q10
- Office Hours: TBA
- Phone: (607)777-4368
- Email: kang@binghamton.edu
- Web: http://www.cs.binghamton.edu/~kang

Teaching Assistants

- Name: TBA
- Email:
- TA Office Hours: TBA
  -

Class Time and Place

- Time: 12pm – 1pm MWF for Section 1 & 1:10pm - 2:10pm MWF for Section 2
- Classroom: SL310 for both sections

Course Objective

This course is designed to provide a solid foundation and background in design and analysis of algorithms. In particular, upon successful completion of this course, you will be able to understand and apply key methodologies for algorithm design and analysis including the following ones:

- Divide and conquer,
- Dynamic programming,
- Greedy algorithms,
• Backtracking,
• Branch and bound,
• Time and space complexity analysis, and
• Theory of NP.
Furthermore, you will be able to develop software by applying these techniques.

Prerequisite

• CS375 (Undergraduate Algorithms and Data Structures) or equivalent
• C/C++ in Linux

Textbook

• Cormen, Leiserson, Rivest and Stein, "Introduction to Algorithm", 3rd (or 4th) edition McGraw-Hill, 2009. (recommended)
• R. Neapolitan and K. Naimipour, "Foundation of Algorithms Using C++ Pseudocode", 3rd (or later) edition, Jones and Bartlett, 2004. (recommended)

Other

The instruction will be primarily based on the instructor’s Lecture Notes that can be downloaded from your course account on blackboard.

Lecture Notes
Lecture Notes for each chapter, in Powerpoint format, will be posted on blackboard before lectures. I recommend you to print lecture notes beforehand and bring them to class so you can take notes easily. Note that lecture notes do not replace class attendance, since (i) they will not be complete and (ii) significant parts of lectures, including in-class discussions and exercises, may not come from the lecture notes.

Main topics

• Introduction
• Time Complexity and Asymptotic Growth Functions
• Divide and Conquer
• Dynamic Programming
• Greedy Algorithms
• Backtracking
• Branch and Bound
• Hashing
• Theory of NP
- Parallel Algorithms (if time permits)
- Linear Programming (if time permits)

---

**Academic Honesty**

Academic honesty and integrity are expected of every student. Dishonesty and cheating, including unauthorized collaboration, in all academic work related to this course will be punished according to the [Watson School Academic Honesty Code](#). If a student is caught for violating the honesty code, he/she will get zero for the corresponding homework, assignment, or exam. In addition, the student must fill in and sign the academic honesty form required by the university.

Students Talk About Academic Honesty: Click [here](#) to watch the video.

---

**Grading policy**

- 5 Homework Assignments: 20% (4% each)
- 4 Projects: 20% (5% each; C/C++ in Linux)
- Midterm: 25% (in the 8th week)
- Final: 30%
- Pop Quizzes: 5% (A pop quiz will be given with no prior notice.)

**Collaboration on Assignments**

Students are encouraged to help one another and to form study groups. In Computer Science, you can sometimes 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 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 assignments individually. Each of you must write your own code, and you must write up all solutions individually. Students submitting solutions 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.

Software plagiarism regarding programming assignments will be checked via [Moss](#). If the programs of two or more students have 80% or higher similarity, they will get zero for the programming assignment. Even if the similarity score is lower than 80, the instructor can require students to prove that they did an assignment on their own if there is any indication or hint of plagiarism.
Some late assignments will be accepted with a late assignment that is 10% per day (including weekends and holidays). No late assignment will be accepted 3 days after the due date.

No makeup exam will be given except for real emergency.

Neither the TA nor the instructor will debug or review your code.

If regrading is requested, the entire assignment or exam will be regraded. As a result, your grade may decrease, increase, or stay the same. The new grade will be final and will not change.

No regrading request will be accepted after two weeks since the grade is assigned. 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.

The instructor will give pop quizzes with no prior notice, and there will be no makeup pop quiz. If you have to miss a class for a valid and verifiable reason, e.g., sickness or a job interview, you have to email the instructor ahead of time.

---

**Attendance and Classroom Policies**

- You are not allowed to use your laptop, notebook, tablet, or smartphone in class unless explicitly permitted. Smartphones must be turned off or in vibrate alert mode during class.
- Attendance is mandatory. The instructor will take attendance frequently. For each absence, your final grade will be reduced by 2 points (out of 100 points that will be used for letter grade assignment). For a late arrival or an early leave (by more than 10 minutes) with no prior approval by the instructor, your final grade will be reduced by 1 point.