CSE551 Foundations of Algorithms

General Information

Time & Location

M/W 10:30AM – 11:45AM @ COORL1-74

Contact Information

Instructor: Baoxin Li  
Office: Brickyard 502  
Office hours: Monday/Thursday 9:00am – 10:00am, plus other times by appointment  
[Cancellation of office hours due to travel or other urgent matters will be announced on Blackboard, and make-up hours will be provided.]
E-mail: Baoxin.Li@asu.edu

Teaching Assistant 1: Lin Chen  
Office: Center Point Tutoring Center Suite 114, Office hours: Tue/Thu 2pm-3pm  
E-mail: Lin.Chen.6@asu.edu

Teaching Assistant 2: Qiongjie Tian  
Office: Center Point Tutoring Center Suite 114, Office hours: Mon/Wed 1-2pm  
E-mail: Qiongjie.Tian@asu.edu

Email Policy

Since email has proliferated, and now constitutes the bulk of extra-classroom conversation between student and instructor, it must be subject to normal rules of formality. Therefore, all email communication will follow the guidelines enumerated here.

✓ Email communication regarding this class MUST include in the subject line the prefix CSE 591: (For example, the subject line of your email may read CSE 591: question about HW1).

✓ Every email must also cc the TAs (unless there is a specific and clear reason why the TA should not be cc’ed). (Note: the TAs are official class staff members and have full access to the Grade Center on Blackboard.)

✓ Emails will be read once a day, M-F. The TAs will directly answer your email, unless the TA (or in some cases, the student) feels that my direct assistance is needed.

✓ Email should be clear, self-contained, and to the point.

✓ Email should not ask questions whose answers are contained in the course syllabus, classnotes/class materials, or other materials on Blackboard.

✓ Students should avoid asking questions in email that should be raised either in class, or in individual consultation with the TA during office hours. These include questions
of an excessively conceptual nature, and questions that require an unreasonable amount of time from the instructor. A good rule of thumb: if you question cannot be answered in a short paragraph, or if it is a question that you should solve on your own through the course of your reading, then it is not appropriate for email.

Emails that do not follow these guidelines will not be replied by the TAs/instructor. If your email goes unanswered one day after you sent it, check if you forgot following these guidelines.

Course Description

Advanced topics in formal algorithm design and analysis, including advanced shortest-paths algorithms, amortized analysis, network flows, NP-completeness, and selected topics in computational geometry, distributed/parallel, randomized, and approximation algorithms.

Course Objective

The goal of this course is to give you advanced knowledge in fundamental algorithms. We will study in depth a set of useful algorithms and learn how they work and why they are considered good. Through this, the course will

- Provide you with solid understanding of these exemplar algorithms so that you have the required theoretic preparation for recognizing situations in which you would be better off looking in the literature or asking experts for a good algorithm to solve your problem instead of just coding the first idea that comes to your mind;
- Give you enough preparation on algorithm analysis so that you are able to understand and navigate the literature on algorithms for choosing proper ones, if they exist, for your problem; and
- Equip you with basic skills for designing new algorithms to solve problems that do not see ready solutions given in the literature.

In order to achieve this, you will have to work through and understand several algorithmic techniques (e.g., divide-and-conquer, dynamic programming, greedy algorithms, network flow algorithms, self-adjusting data structures, basic randomized/approximation techniques) and the mathematical background necessary for analyzing the properties of these techniques and the algorithms based on them (e.g., amortized analysis, recurrence relations, basic graph theory, NP-completeness).

The students are expected to understand the material typically covered in a junior level CS Data Structures and Algorithms class (e.g., CSE310 at ASU) and its prerequisites such as Discrete Mathematics (e.g., MAT243 at ASU). In particular, you should know how quicksort and mergesort work (and be able to write and solve recurrence relations for those), you should be able to use the "big-Oh" notation and you should have seen the algorithms of Dijkstra and Prim and have at least an intuitive understanding of how they work. We will also assume you know the definitions and basic properties of heaps and binary search trees.

As this is a graduate-level CS course, obviously we assume you know computer programming.

This is intended to be a rigorous, theoretical course. In particular, the evaluation will be mostly based on in-classroom exams (more later). All exams will be closed-book.
Pre-requisite: Graduate Standing in Computer Science and Engineering

Textbooks


The following book is recommended as additional reference:


Blackboard on my.asu.edu

Course materials (notes, assignments, solutions, important announcements etc.) will be posted on Blackboard. It is your responsibility to keep updated by checking Blackboard regularly. (For example, if an announcement was made on Blackboard regarding the exact date/time of an exam but you did not check there and thus missed the exam, you cannot be excused by simply arguing “I did not know; you didn’t tell us about the exam in the class”).

Topics to Cover

We will cover most of chapters 1, 4, 5, 6, and 7, and selected topics from chapters 8, 11, and 13. Note that some of the topics in chapters 4, 5, and 6 have already been covered in CSE 310, so we will focus on the new material in these chapters. We may also cover a few selected topics which are not covered by the textbook (e.g., splay trees, skip lists, amortized analysis).

Below is a tentative schedule of topics:

- Stable Matching (and 5 representative problems, Section 1.2) (1 week)
- Greedy Algorithms (2.5 weeks)
  * Interval Scheduling
  * Interval Partitioning
  * Min Lateness Scheduling
  * Caching
  * Cycle and Cut Properties for Min Spanning Trees
  * Shortest Paths: proof of correctness of Dijkstra
  * Huffman codes
- Amortized Analysis (1 week)
  * log^* n analysis on Union-Find Data structure
  (and basic knowledge about the data structure)
- Splay Trees (1 week)
- Divide-and-Conquer (1.5 weeks)
  * Intro Computational Geometry: Closest Pair of Points
  * Fast Integer and Matrix Multiplication
    (Karatsuba's and Strassen's algorithms)
Dynamic Programming (2 weeks)
* Weighted Interval Scheduling
* Knapsack: pseudo-polynomial algorithm
* Shortest Paths: Bellman-Ford
* RNA secondary structure

Distributed algorithms: Distance Vector Protocols (0.5 weeks)

Network Flows (2 weeks)
* Ford-Fulkerson Algorithm
* max flow-min cut theorem
* capacity scaling algorithm/Edmonds-Karp algorithm
* max cardinality bipartite matching
* max number of edge-disjoint paths

Polynomial Time Reductions and NP-completeness (2 weeks)
* classes P and NP
* NP-completeness

Approximation algorithms (1 week)
* Traveling Salesman Problem: simple 2-approximation based on Min Sp. Trees
* Load Balancing
* Knapsack
* other approximation algorithms, time permitting

Randomization: Skip Lists (0.5 week)

**Lecture Notes:** The lecture notes from the publisher will be utilized, with slight modifications whenever needed. In addition, some examples may be worked out on the whiteboard during the lectures. The examples may not be included into the posted lecture notes.

**Grading Policy**

Your grade will be based on homework assignments and exams (three exams, including the final). All exams will be closed-book.

Homework: 25% (There will be about 7~8 assignments. Some of the homework assignment may involve programming.)

Midterm exams (two; time TBD): 25% each

Final Exam: 25%

The following is the most likely cutoff for the final letter grades (adjustments, if any at all, should be minute):

A: >= 86%  B: >= 71%  C: >= 60%  D: >= 50  E: < 50

**Note:** +/- will be linearly interpolated, except that A+ is rarely given.
On Homework:

Homework assignments will be graded only “coarsely”, and you will get the respective credits for the problems if your answers show you have tried earnestly. We will post the solutions.

Submission of homework must be electronic. You can either type your work or take pictures of your hand-written sheets.

Late submissions will not be considered. (Genuine emergencies such as medical emergencies that prevent on-time submission may be considered on a case-by-case basis, provided that official documents are supplied to verify such emergencies.)

Brief summary of the University policies on cheating

Academic Integrity
All students in this class are subject to ASU’s Academic Integrity Policy (available at http://provost.asu.edu/academicintegrity) and should acquaint themselves with its content and requirements, including a strict prohibition against plagiarism. All violations will be reported to the Dean’s office, who maintain records of all offenses. Students are expected to abide by the FSE Honor Code (http://engineering.asu.edu/integrity/).

Any incidence of cheating in this class will be severely dealt with. This applies to homework assignments, programming assignments, quizzes and tests. The minimum penalty for cheating will be that the student will not obtain any credit for that particular assignment. (This means that if in a test or assignment a student is found to have cheated, he/she will obtain zero in that test/assignment.) For the homework assignments students are encouraged to discuss the problems with others, but one is expected to turn in the results of one's own effort (not the results of a friend's efforts). One tends to get very suspicious if two identically wrong results show up in the homework assignment and/or tests. The names of the offenders will be maintained in the departmental files. The repeat offenders may be debarred from the University. Note: the instructor is obliged to reporting academic dishonesty to the dean’s office.

We will assign your seat during any exam; You cannot choose your seat.

Absence & Make-Up Policies
No make-up exam will be given unless the absence is due to genuine emergencies that are backed up by official documents. Accommodations will be made for religious observances provided that students notify the instructor at the beginning of the semester concerning those dates. Students who expect to miss class due to officially university-sanctioned activities should inform the instructor early in the semester. Alternative arrangements will generally be made for any examinations and other graded in-class work affected by such absences.

Classroom Behavior
Cell phones and pagers (must be/or state alternative rule) turned off during class to avoid causing distractions. The use of recording devices (is/is not) permitted during class. Any violent or threatening conduct by an ASU student in this class will be reported to the ASU Police Department and the Office of the Dean of Students.
Disability Accommodations
Suitable accommodations will be made for students having disabilities and students should notify the instructor as early as possible if they will require same. Such students must be registered with the Disability Resource Center and provide documentation to that effect.