CSCE-629 Analysis of Algorithms

Spring 2019

Instructor: Dr. Jianer Chen
Office: HRBB 315C
Phone: 845-4259
Email: chen@cse.tamu.edu
Office Hours: T,Th 11:00 am–12:30 pm

Teaching Assistant: Qin Huang
Office: HRBB 315A
Phone: 402-6216
Email: huangqin@email.tamu.edu
Office Hours: MWF 3:00 pm–4:00 pm

Algorithm design and analysis is a fundamental and important part of computer science and engineering. This course introduces students to advanced techniques for the design and analysis of algorithms, and explores a variety of applications.

The course is a *continuation* of the undergraduate algorithm analysis course (such as CSCE-411). The objective of this course is to study and analyze a broad variety of important and useful algorithms: methods for solving problems which are suited for computer implementation. We will deal with many different areas of applications, always trying to concentrate on "fundamental" algorithms that are important to know and interesting to study. We will also study the "complexity", i.e., the inherent difficulties of practical problems. The topics that will be discussed in this course include: advanced data structures, advanced design and analysis techniques, advanced graph algorithms, NP-completeness theory, approximation algorithms, randomized algorithms, and parameterized algorithms.

We will *not* follow the textbook chapter and chapter. Instead, we will select proper materials from the textbook, starting with a brief review of the basic definitions in algorithm analysis. Supplementary lecture notes will be provided for materials not included in the textbook. A more detailed description for the course can be found in page 2. There will be six homework assignments plus a programming project. The assignments are due on the designated due dates *at the beginning of class*. **No late submissions will be accepted**. Discuss unusual circumstances *in advance* with the instructor.

A midterm examination will be given on Thursday, March 7, from 12:45 pm to 2:00 pm. The final examination is on Tuesday, May 7, from 8:00 am to 10:00 am.

Textbook: T. Cormen, C. Leiserson, R. Rivest, and C. Stein *Introduction to Algorithms*, 3rd edition, MIT Press and McGraw-Hill, Cambridge, MA, 2009.

The following books are also useful and helpful.

1. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979, and

2. J. Kleinberg and E. Tardos, *Algorithm Design* , Addison Wesley, Boston, 2006.

Supplementary lecture notes will also be provided.

Grading: Homework 30%, Programming 10%, Midterm 20%, and final exam 40%.

Remark. Course materials, lecture notes, and homework assignments can be found via the instructor's home page: http://faculty.cse.tamu.edu/chen

What am I supposed to know before taking this course

Background of an undergraduate algorithm course. I will be assuming that you are familiar with the topics listed below. Discussion on these materials can be found in our textbook. Some of the materials here may be quickly reviewed in class.

- pseudo-code for describing algorithms
- big-Oh notation and complexity analysis
- related mathematics foundations
- data structures such as lists, stacks, queues, and binary trees
- sorting: insertion sort, selection sort, mergesort, quicksort, heapsort, radix sort
- basic graph concepts and representations
- depth-first search and breadth-first search
- Dijkstra shortest path algorithm
- Dijkstra-Prim minimum spanning tree algorithm
- the algorithm for testing biconnectivity
- Strassen's matrix multiplication algorithm
- Warshall's transitive closure algorithm
- general idea of P, NP, and NP-completeness

What am I going to study in this course

- search tree structures (worst case analysis)
- union and find (amortized analysis)
- topological sort (application of DFS)
- strongly connected components (application of DFS)
- spanning tree algorithms (greedy algorithms, Kruskal's and Prim's approaches)
- single-source path algorithms (Dijkstra's algorithm and Bellman-Ford algorithm)
- decrease-and-conquer (selection problem and its applications)
- maximum flow algorithms (Ford-Fulkerson's algorithm and Dinic Algorithm)
- graph matching algorithms
- dynamic programming (string and sequence algorithms)
- NP-completeness theory
- approximation algorithms
- randomized algorithms
- parameterized algorithms
- bigdata algorithms