Instructor: Yin Yang, yin5@clemson.edu

Prerequisites: Data Structure & Algorithm Analysis, or instructor permission.

Textbook: *T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Introduction to Algorithms, 2nd Edition, McGraw-Hill, 2001.*

Office hours: Monday morning 8:00 – 9:00 AM or by appointment (virtual on Zoom)

Grading:

There will be three homeworks which will make 40% of the final grade. In addition, we also have three exams (60% in total):

Exam I (in late Feb.): 20%

Exam II (in late Mar.): 20%

Exam III (in late Apr.): 20%

Catalog description:

The course is primarily theoretical in nature, with a focus on understanding fundamental aspects of algorithms and computing. In the meantime, I will put concrete examples in engineering/numerical computing to illustrate how such abstract concepts can be used in CS and computing fields. The main activity associated with exams will be analysis. We will consider various models of computation, the limits of what can and cannot be computed using these models, as well as what problems can be solved efficiently using these models. We will also study a number of important techniques that can be used to improve the efficiency of algorithms. There will be very few (if any) programming activities in this class; however, you may find tools such as Matlab useful in your analyses, and you may use the programming language of your choice to validate solutions.

Tentative Course Topics:

• Introduction to the Course

Objective: Introduction to the course and course administration.

• Proof Techniques

Objective: To review the basic ideas and logic behind mathematical proofs. Construction, induction, contrapositive, and contradiction techniques will be considered.

• Formal Language Theory & Models of Computation

Objective: To present basic models of computation, determine their limitations, and establish basic computational complexity classes. Deterministic and Nondeterministic Automata, Mealy and Moore Machines Turing Machines, RAM Machines, Church-Turing Thesis, Quantum Computers.

• Algorithmic Analysis

Objective: A review of the analysis of algorithms, followed by a review of divide-and-conquer algorithms. Divide-and-conquer recurrences and their solution using the characteristic equation method as well as generating functions.

• Algorithmic Techniques

Objective: A discussion of divide-and-conquer algorithms, dynamic programming, and greedy algorithms, along with consideration of the problem elements that should exist in order for any of these techniques to be applicable.

• Amortized Analysis

Objective: Review of worst- and average-case analysis, followed by a presentation of amortized analysis, and how it relates to worst- and average-case analysis.

• Advanced Data Structures

Objective: To study advanced data structures using amortized analysis.

• Graph Algorithms

Objective: Review of basic graph algorithms, followed by a consideration of the minimal spanning tree, shortest paths, and network flow algorithms.

• NP-Completeness (advanced topics)

Objective: To provide a brief overview of computational complexity theory, and the tools that are necessary to prove a computational problem NP-complete.

• Linear Programming (advanced topics)

Objective: To study optimization algorithms, with a focus on linear programming, and to apply these to the study of game theory.