Design and Analysis of Algorithms (CS 07.340)

Catalog Description

In this course, students will learn to design and analyze efficient algorithms for sorting, searching, graphs, sets, matrices, and other applications. Students will also learn to recognize and prove NP-completeness.

Instructor

Name: Dr. G. Baliga
Email: baliga@rowan.edu
Office hours: Monday and Wednesday from 2-3:30pm

Text

Anany Levitin, *Introduction to the Design and Analysis of Algorithms*, Pearson publishers.

Objectives

This course focuses on advanced problem-solving techniques including divide and conquer, dynamic programming, greedy design, backtracking, branch and bound and probabilistic algorithms. Advanced data structuring techniques will be studied within the context of designing efficient graph algorithms. Students will gain an appreciation for the fundamental notion of NP-completeness.

Major Topics

The topics in this course are taken from the following:

- Mathematical Preliminaries
  - Big-O notation for algorithm complexity
  - Proof by mathematical induction
  - Proof by contradiction
- Divide and Conquer
  - Large precision integer arithmetic
  - Advanced sorting algorithms
- Dynamic Programming
  - Efficient chain matrix multiplication
  - Optimal binary search trees
  - All-pairs shortest path problem in a graph
- Greedy Design
  - Prim’s and Kruskal’s Minimum Spanning Trees (MST) algorithms
  - Dijkstra’s shortest path algorithm in a graph
  - Application of the greedy design technique to problems arising within other domains such as geometry, scheduling, computer networks etc.
- Backtracking

    - Depth first search
    - Applications of backtracking for problems chosen from
        - Boolean Formula Satisfiability (SAT)
        - N-queens on a NxN chessboard
        - Graph coloring problem
        - Traveling salesman problem
- Branch and Bound
    - Breadth first search
    - Breadth first search with branch and bound pruning
    - Best first search with branch and bound pruning
    - Applications of branch and bound for problems such as the 0-1 knapsack problem.
- Advanced Data Structures
    - Heaps
    - AVL trees or Red Black trees
    - Union-Find ADT implementation
- NP-completeness
    - Polynomial time problem reduction
    - Structure of NP-completeness proof. One or more of the following problems may be used to illustrate proof of NP-completeness.
        - Hamiltonian Path
        - Hamiltonian Circuit
        - Traveling Salesman
        - Graph 3-Coloring
        - Satisfiability (SAT)
        - Clique
        - Independent Set
        - Vertex Cover

Grading

Grades will be based on performance in the following components. The relative weights of the components are as given below:

Quizzes/Homework	70%
Final Exam	15%
Attendance/Participation	15%

Unless specified otherwise, all quizzes are weighted equally. Quizzes may be held unannounced. A re-quiz for a missed quiz may be provided at the instructor's discretion and only when there are documented, extenuating circumstances. The final exam will be held on the scheduled time during finals week. Attendance is mandatory and is taken at the beginning of each lecture.

The correspondence between numerical score and letter grade is as follows: 93+ (A), 89+ to 93 (A-), 86+ to 89 (B+), 83+ to 86 (B), 80+ to 83 (B-), 77+ to 80 (C+), 74+ to 77 (C), 71+ to 74 (C-), 67+ to 71 (D+), 63+ to 67 (D), 59 to 63 (D-), less than 59 (F).