Design and Analysis of Algorithms

MET CS 566

Instructor: Dr. Madani Naidjate
Email: *madani@bu.edu*
Office hours: Before and after each class meeting.

Course Description:

Algorithms are the soul of computing. Algorithmic thinking, unlike the very young electronic machinery it brings alive, is rooted in ancient mathematics. It can be roughly described as creating *"recipes"* (well defined sequences of computational steps) for getting *"things"* (computational problems specifying an input-output relation) *"successfully"* (correctly) *"done"* (in finite steps and time). This course introduces basic methods for the design and analysis of efficient algorithms emphasizing methods useful in practice. Different algorithms for a given computational task are presented and their relative merits evaluated based on performance measures. The following important computational problems will be discussed: sorting, searching, elements of dynamic programming and greedy algorithms, advanced data structures, graph algorithms (shortest path, spanning trees, tree traversals), string matching, elements of computational geometry, NP completeness.

Course Objectives

Algorithm design and analysis provide the theoretical backbone of computer science and are a must in the daily work of the successful programmer. The goal of this course is to provide a solid background in the design and analysis of the major classes of algorithms. At the end of the course students will be able to develop their own versions for a given computational task and to compare and contrast their performance.

Prerequisites

METCS248 (Discrete Mathematics) or equivalent, and CS341 or CS342 (Data Structures) or instructor's consent

Grading

New material will be presented in lecture format. Reviews, exercises and homework solutions will take place in discussion. Participation in the discussions, although not mandatory, is recommended and may result in extra credit. Homework is due one week after handed out. Late homework will not be accepted unless permission by the instructor was given prior to the due date. No predetermined scale will be used. The final grade will be assigned based on the following weighting:

Assignments: 20% *Midterm:* 30% *Final:* 30% *Presentation:* 20%

Textbook

1. Lecture Notes. Available on-line.

2. T. H. Cormen, C. E. Leiserson, R. L. Rivest. Introduction to Algorithms The MIT Press, Cambridge, Massachusetts, 3rd edition. (Recommended not required) Available on PDFormat on line.

Courseware:

Course materials for this class can be found at: *people.bu.edu/madani.* Assignments solutions and announcements will be available on Blackboard.

Class Policies

1) **Attendance & Absences** – Students are expected to attend and sit through the entire class meetings. In case of an absence, the student is responsible to arrange for notes and missed announcements. Three (3) or more absences may result in a withdrawal from the class.
2) **Assignment Completion & Late Work** – Late assignment submission is not allowed, unless a permission is granted by the instructor prior to the deadline. Students should submit a physical to the instructor by the due date. If the student is to be absent that day, completed assignments should be submitted via email before class begins.
3) **Academic Conduct Code** – Cheating and plagiarism will not be tolerated in any Metropolitan College course. They will result in no credit for the assignment or examination and may lead to disciplinary actions. Please take the time to review the Student Academic Conduct Code:

   http://www.bu.edu/met/metropolitan_college_people/student/resources/conduct/code.html.

   NOTE: [This should not be understood as a discouragement for discussing the material or your particular approach to a problem with other students in the class. On the contrary – you should share your thoughts, questions and solutions. Naturally, if you choose to work in a group, you will be expected to come up with more than one and highly original solutions rather than the same mistakes.]

Course Outline:

1. Introduction
   - What is an algorithm?
   - Notation for programs
   - Proof techniques
   - Basics review: Sets - Functions - Limits - Simple series
2. Fundamentals
   - Instances and problems - Elementary operations.
   - Efficiency
   - Average and worst-case analysis
   - Examples
3. Asymptotic notation
   - Introduction

    - A notation for "the order of"
    - The omega notation
    - The theta notation
    - The conditional asymptotic notation
4. Analysis of algorithms
    - Analyzing control structures
    - Using a barometer
    - Amortized analysis
    - Solving recurrences
5. Data structures
    - Arrays, stacks and queues
    - Records and pointers
    - Lists, graphs, trees and associative tables
    - Heaps
    - Disjoint set structures
6. Greedy algorithms
    - Making change
    - General characteristics of Greedy algorithms
    - Graphs MST - Kruskal's and Prims's algorithms
    - Graphs: shortest paths
    - Knapsack problem
    - Scheduling
7. Divide - and - Conquer
    - Multiplying large integers
    - Binary search
    - Sorting by: merging and quicksort
    - Finding the median
    - Matrix multiplication
    - Exponentiation
    - Quick look at cryptography
8. Dynamic programming
    - Making change
    - Principles of optimality
    - The knapsack problem
    - Shortest paths - Floyd's algorithm
    - Chained matrix multiplication
9. Introduction to probabilistic algorithms - Parallel algorithms
10. Introduction to computational complexity