CS-7200-02 Algorithm Design and Analysis

- **Instructor:** Cemil Kirbas
- **Phone No.:** (937)-775-1019
- **Email:** cemil.kirbas@wright.edu
- **Quarter:** Fall, 2015
- **Class Hrs:** MW, 6:10-7:30pm, Brehm Lab 165
- **Office Hrs:**MW, 7:45-8:30pm (437 RC)
- **GTA and Office Hours**: TBA
- **GTA and Study Table Helper:** TBA

Course Objectives

To provide a solid foundation in algorithm design and analysis. Specifically, the student learning outcomes include:

- Basic knowledge of graph and matching algorithms.
- Ability to understand and design algorithms using greedy strategy, divide and conquer approach, dynamic programming, and max flow - min cut theory.
- Ability to analyze asymptotic runtime complexity of algorithms including formulating recurrence relations.
- Basic knowledge of computational complexity, approximation and randomized algorithms.

Prerequisites

CS 3100/5100 Data Structures and Algorithms

Course Description

This course introduces concepts related to the design and analysis of algorithms. Specifically, it discusses recurrence relations, and illustrates their role in asymptotic and probabilistic analysis of algorithms. It covers in detail greedy strategies, divide and conquer techniques, dynamic programming and max flow - min cut theory for designing algorithms, and illustrates them using a number of well-known problems and applications. It also covers popular graph and matching algorithms, and basics of randomized algorithms and computational complexity. However, the depth of coverage of complexity classes and intractability, approximation algorithms, and randomized algorithms, will be as time permits. The programming assignments can be coded in MATLAB, Python or Java.

Course Load

The course load includes homeworks and programming assignments, a midterm exam, and a final exam.

Policies:
(1) You are encouraged to work in teams and share your ideas with each other to improve your understanding of the material through discussions of the lecture material and assignments. The assignments will be graded. Even though you can collaborate and discuss the assignments, you should write up solutions on your own, in your own words.

1) The exams will be closed notes and closed book. No cheat sheets will be allowed in the exam. Instead, the exam will provide helpful information and code snippets from the text book or course notes that are deemed necessary for answering questions.
2) The students will be prohibited from using any computer (e.g., laptop), calculator, or cell phone during the exam.
3) Any academic impropriety during an exam (which includes copying from other students, or accessing resources using electronic devices which are prohibited, and passing it off as your own work) will have a minimum penalty of an 'F' grade, plus additional disciplinary action for unethical behavior. See http://www.wright.edu/students/judicial/integrity.html for details.

The entire course grade will be determined using the scores on the homeworks worth 20 points and two proctored exams: a midterm exam worth 40 points, and a final exam worth 40 points.

Required Text

1. J. Klienberg and Eva Tardos: Algorithm Design. 1st Edition. Addison Wesley, 2006. ISBN 0-321-29535-8

Reference Text

1. T. H. Corman, C. E. Leiserson, R. L. Rivest, and C. Stein: Introduction to Algorithms. 3rd Edition. MIT Press, 2009. ISBN 978-0-262-53305-8

Grading

The letter grades will be assigned using the following scale: A[90-100], B[80-90), C[70-80), D[60-70), and F[0-60). However, I reserve the right to adjust the scale somewhat to utilize the gaps in the distribution.

Topics

Class 1	Introduction; Stable Matching Problem
Class 2	Gale Shapley Algorithm
Class 3	Algorithm Analysis; Graph Search
Class 4	Greedy Algorithms : Interval Scheduling/Partitioning
Class 5	Shortest Paths Algorithm
Class 6	Minimum Spanning Trees; Huffman Code
Class 7	Divide and Conquer : Inversions, Closest Pair of Points
Class 8	Recurrence Relations and Master Theorem
Class 9	Divide and Conquer : Multiplication
Class 10	Fast Fourier Transform
Class 11	Quicksort : Average Case Analysis; Randomization
Class 12	Select / Median Algorithm : Probabilistic Analysis
Class 13	Dynamic Programming : Weighted Intervals
Class 14	Segmented Least Squares; Knapsack
Class 15	RNA Secondary Structure; Sequence Alignment
	Midterm Exam
Class 16	Sequence Alignment in Linear-space
Class 17	Bellman-Ford Algorithm
Class 18	(continue)
Class 19	Maximum Flow Theory: Ford-Fulkerson Algorithm
Class 20	Maximum Flow Applications
Class 21	Intractability: Polynomial-Time Reduction
Class22	NP-Completeness: Definitions and Proofs
Class 23	Complexity classes
Class 24	PSPACE
Class 25	Extending Limits of Tractability
Class 26	Approximation Algorithms
Class 27	Local Search
Class 28	Randomized Algorithms
	Final Exam