ALGORITHMS IN THE REAL WORLD

1. Catalog Description

The course seeks to raise the interest in theoretical foundations of computer science among senior level undergraduates by examining how popular applications (such as Google, Cryptography, Data Compression, Web Crawling, Mapquest) work. Most of these applications involve significant advances in algorithms and data structures, and this course will examine the theory underlying them. Rigorous design and analysis of algorithms will be emphasized, along with attention to modeling of complex problems combinatorially. The applications are chosen so as to illustrate as many different algorithmic paradigms as possible. Prerequisites: (i) CS 2604 or CS 2606, (ii) MATH 3034 or MATH 3134.

2. Learning Objectives

Having completed this course, students would be able to

(1) Understand different algorithmic paradigms and the power of different kinds of data structures designed for specific problems,
(2) Analyze algorithms and heuristics rigorously,
(3) Understand the mathematics and algorithms underlying some of these well known applications, and appreciate how they are designed.
(4) Develop abstract combinatorial models for a complex problem,
(5) Develop problem solving skills.
(6) Have a foundation for further theoretical work in computer science.

3. Justification

A number of recent applications (e.g. Google, Cryptography, Data Compression, Web Crawling, Mapquest, Routing Protocols, Akamai and Amazon), primarily motivated by the Internet, are based on significant algorithmic innovations. The main motivation of this course is to look closely at some of these popular applications and identify and understand the core algorithmic ingredients. The goal is to make problem solving, algorithms and data structures more realistic and down-to-earth for students, and raise interest in theoretical foundations of computer science.

The algorithms we will discuss are not necessarily exactly what these applications use, but will serve to illustrate the general concepts. For instance, Google News uses a clustering technique to classify news headlines, according to its website, but the specific clustering technique seems to be proprietary, and instead we will just discuss hierarchical clustering. The applications, problems and algorithms are chosen so that (i) students have heard of them or have used them, and (ii) they can be used to illustrate as many different algorithmic design paradigms as possible, and highlight the utility of algorithms in everyday applications.

4. Prerequisites and Corequisites

The specific concepts required are:

(1) Basic discrete mathematics and combinatorics,
(2) Exposure to analysis of algorithms, $O()$ and $\Omega()$ notation,
(3) Basic graph theory: graphs, trees, paths,
(4) Basic linear algebra: vectors, matrices, eigenvalues,

(5) Basic probability theory: random variables, expectation, independence,
(6) Basic programming skills,
(7) a practical understanding of abstract data structures and their use in algorithm design and analysis,
(8) ability to think abstractly and use standard proof techniques.

The above prerequisites seem to be adequately covered by the courses (i) CS 2604 or CS 2606, and (ii) Math 3134 or 3034.

5. Texts and Special Teaching Aids

The course would be based on:

(1) J. Kleinberg and E. Tardos. Algorithm Design, Addison Wesley, ISBN 0-321-29535-8.
(2) S. Dasgupta, C. Papadimitriou and U. Vazirani. Algorithms, McGraw Hill Publishers. A draft is available at *http://www.cse.ucsd.edu/users/dasgupta/mcgrawhill/*
(3) K. Bryan and T. Leise. The $25,000,000,000 eigenvector. The linear algebra behind Google. *SIAM Review*, 48 (3), 569-81. 2006. Also avaiable at *http://www.rose-hulman.edu/ bryan/google.html* to think abstractly and use standard proof techniques.
(4) Lecture notes

6. Syllabus

The course consists of three modules: Graph Theory, Number Theory and Linear Algebra, and we will discuss algorithmic problems in each of these modules. The topics include:

(1) Graph Search algorithms: review of basic graph theory and algorithm analysis, graph search algorithms, Web Crawling
Percent of Course: 5
(2) Shortest Paths: Dijkstra and Bellman-Ford algorithms, review of heaps
Percent of Course: 10
(3) Cryptography: binary arithmetic, modular exponentiation, primality (a restricted version of Rabin-Miller), and the RSA public key system.
Percent of Course: 25
(4) Spanning trees and hierarchical clustering: Greedy algorithms for spanning trees, union-find data structure, hierarchical clustering, applications to Google News
Percent of Course: 15
(5) Data Compression: Huffman's algorithm, and a brief discussion of Lempel-Ziv and JPEG.
Percent of Course: 10
(6) Network Flows: Introduction to flows and cuts, duality theorem, Ford-Fulkerson and Capacity scaling algorithms, applications to scheduling and assignment problems
Percent of Course: 25
(7) Analysis of social networks: review of linear algebra, Google's PageRank algorithm.
Percent of Course: 10

7. Old (Current) Syllabus

Not Applicable

8. Core Curriculum Guidelines