

CS4234/CS5234 : Combinatorial and Graph Algorithms

CS4234/CS5234 : Combinatorial and Graph Algorithms

Fall 2001 (Jul-Oct 2001)
Leong Hon Wai, S16 05-05

Pre-requisites: CS3230 Design and Analysis of Algorithms

Assessment: (25%) Homework
(15%) Programming Project -- using C++ and LEDA
(60%) Final Exam [Open Book]

[Graduate and Undergraduate students will be graded separately.
Graduate students will do more work in the course.]

Objectives:

This course presents *advanced* material on the design and analysis of combinatorial algorithms with emphasis on efficient algorithms and data structures. It also provides practical experience of re-using a C++ library of advanced data structures and algorithms in software development. This course is meant for students who intend to do (i) research work in computer science in general, and algorithm design in particular, or (ii) advanced application/software development in other areas of computer science.

Tentative Course Outline (Fall 2001)

A. ADVANCED DATA STRUCTURES

- Priority Queues, Heaps and Graph Algorithms
- Leftist Heaps and Fast MST Algorithm
- Amortized Analysis and Binomial Heaps (fast review)
- Fibonacci Heaps and Fast Shortest Path Algorithm
- LEDA - A Library of Efficient Data Structures and Algorithms

B. COMBINATORIAL ALGORITHMS

- Maximum Matching -- Problems, Algorithms, Applications
- Network Flow -- Problem, Algorithms, Applications
- Knapsack Problem -- Problem, Algorithms, Applications
- Graph Partitioning -- Problem, Algorithms, Applications
- Berth Allocation Problem

C. NP-COMPLETENESS

- Cook's Theorem
- Proving NP-Completeness
- Approximation Algorithms
- Local Search Methods

Project Work:

For project work, each student is expected to do a C++ programming project using LEDA to solve a combinatorial optimization problem -- to be announced later. LEDA is a powerful C++ library of data structures and algorithms that can be reused to

CS4234/CS5234 : Combinatorial and Graph Algorithms

implement advanced algorithms covered in the course. To help learn LEDA and realize the power of LEDA, one or two simpler implementation exercises will also be included.