Tentative Syllabus
CSE737, Advanced Topics in Algorithms Design and Analysis
Spring Semester, 2002

Class time: 6:00–9:00 p.m., Wed.
Building & Room: Baldy 45
Professor-in-Charge: Jinhui Xu
212 Bell Hall
Phone: 645-3180 ext. 132
E-mail: jinhui@cse.buffalo.edu
Office Hours: 4:00 – 5:00 p.m., Thu.

General Description: The is a research oriented seminar course which covers several important topics in algorithm design and analysis. There are several objectives for this course: (1) Introduce students to these topics and their recent developments. (2) Teach them the underlying methodologies and techniques which will be benefitial to their own researches. (3) Accumulate students' research experiences. The classes will be a mix of lectures and student presentations. Each student will assign several papers and need to present them to the rest of the class.

Prerequisites:

Courses: CSE531, and CSE632 (strongly recommended)

Topics: A high level language such as C, C++, or Java, and familiar with Latex.

Text: There is no required textbook for this course. Following are some reference books.

- Vijay V. Vazirani, *Approximation Algorithms*, Springer, 2001.
- D. S. Hochbaum (editor), *Approximation Algorithms for NP-Hard Problems*, PWS Publishing Company, 1997.
- Motwani and Raghavan, *Randomized Algorithms.* Cambridge University Press, 1995.

Topics to Be Covered (Tentative)

- Advanced techniques: Fractional cascading, parametric search, prim-dual method.
- Data structures: Range-tree, Suffix tree, integer data structure, kinetic data structure
- Approximation algorithms: Euclidean TSP, Multi-way cut, Voronoi Diagram and Nearest Neighbor, Steiner Tree, Scheduling, Bin Packing, Knapsack.

Course Work and Grading:

Paper Reading and Presentations: 40%

Notes preparation and class reports: 25%

Project : 30%

Class Attendance and Discussion: 5%

Each student will read several papers which form a topic, and need to present them in class. Each student has a project which is either write a survery on the selected topic, implement an important algorithm in that topic, or improve an existing algorithm.

This course will only give S/U grade. In order to get an S, students need to get 60% of the total scores.