COURSE DESCRIPTION

Department and Course Number: COMP 4270
Course Title: Advanced Algorithms
Total Credits: 3
Required: No
Prerequisites: COMP 3270
Class meetings per week: 3 hours
Lab meetings per week: 0 hours
Course Coordinator: Dr. N. Hari Narayanan
Date Prepared: February 20, 2004

Current Catalog Description:
Fundamentals of designing and analyzing advanced algorithms. Algorithm design theory; computational complexity; relationship of data structures to algorithm design; study of design strategies including divide-and-conquer, the greedy method and dynamic programming.

Textbooks:

References:

Course Objectives:
1. Learn advanced algorithms for networks, matrices, polynomials, number theory, strings and geometry.
2. Be able to analyze the complexity of these algorithms.
3. Be able to apply advanced algorithm design techniques.
4. Develop an understanding of NP-completeness and related topics.

Prerequisites by Topic:
1. Knowledge of basic algorithm analysis and proof techniques
2. Familiarity with a variety of fundamental data structures and algorithms

Topics Covered: (specify number of hours on each)
1. Mathematics: asymptotic notation, summations and recurrences (3 hours)
2. Divide-and-Conquer and Greedy algorithms (4 hours)
3. Dynamic programming (4 hours)
4. Flow networks (4 hours)
5. Matrix operations (4 hours)
6. Polynomials and FFT (3 hours)
7. Number-theoretic algorithms (4 hours)
8. String matching (4 hours)
9. Computational geometry (3 hours)
10. NP-completeness (4 hours)
11. Approximation algorithms (4 hours)
12. Exams (4 hours)

**Laboratory Projects:** (specify number of weeks on each)
1. Divide and conquer algorithms (2 weeks)
2. Greedy algorithms (2 weeks)
3. Dynamic programming (2 weeks)
4. Approximation algorithms (2 weeks)

**Oral and Written Communications:**
None.

**Social and Ethical Issues:**
None.

**Theoretical Content:**
Mathematical techniques for proving the correctness of algorithms and analyzing their complexity are presented and applied throughout this course.

**Problem Analysis and Solution Design:**
All students apply techniques to analyze, design, implement, and test algorithms in their problem solving and programming assignments.