

CS6505: Computability, Algorithms, and Complexity

Fall 2003

Description: This course will cover: (a) important concepts from computability theory; (b) techniques for designing efficient algorithms for combinatorial, algebraic, and number-theoretic problems; and (c) basic concepts such as NP-Completeness from computational complexity theory.

This course can be taken for satisfying the theory breadth requirement by graduate students, especially non-theory Ph.D. students. This course cannot be taken by theory/ACO Ph.D. students to satisfy the breadth/core requirement. Students who are interested in theory and are well prepared are advised to take cs6550 rather than this course.

Pre-requisites: cs3500 or equivalent.

Text books:

- (a) Sipser: Introduction to the Theory of Computation
- (b) Cormen, Leiserson, Rivest, and Stein: Introduction to Algorithms

Assignments and Tests:

Homeworks: 40%, mid-term: 25%, final exam: 35%.

Topics:

1. Computability: The Halting Problem, Reducibility, Undecidable Problems from Language Theory.
2. Algorithms and complexity:
 - (a) Dynamic programming
 - (b) Minimum spanning trees using Fibonacci heaps, UNION-FIND
 - (c) Shortest Path (Dijkstra, Bellman-Ford), Floyd-Warshall algorithm
 - (d) Max-flow
 - (e) Matching
 - (f) Algebraic algorithms: modular exponentiation, FFT
 - (g) Basic Randomized algorithms
 - (h) Basic cryptography, RSA
 - (i) Basic notions from complexity theory, NP-Completeness
 - (j) Basic approximation algorithms
 - (k) Others as time permits