COMP 3270, Introduction to Algorithms

Credit hours: 3 lecture
Contact hours: 3 lecture

Catalog Description: Algorithms for standard computational problems and techniques for analyzing their efficiency; designing efficient algorithms and experimentally evaluating their performance.

Prerequisites: COMP 2210 and COMP 3240
Corequisites: None

Required Course (CSCI, ECPE, SWEN, WIRS)

Instructor or Course Coordinator: Dr. Hari Narayanan

Required Textbook

Course Outcomes
The student will be able to
- understand fundamental algorithms for data structure operations and solving various problems.
- reason about how algorithms operate on specific inputs.
- determine and prove whether an algorithm is correct.
- analyze algorithms and calculate their efficiency.
- debug algorithms.
- modify and reuse algorithms.
- learn different algorithm design techniques.

Topics Covered
- Introduction: computational thinking and problem solving (1.5 hours)
- Designing, understanding and analyzing algorithms: pseudocode, correctness and complexity of algorithms (1.5 hours)
- Proof techniques for algorithm correctness: counterexample, contradiction, loop invariants and induction (3 hours)
- Growth functions and notations: big-oh, omega, small-oh, small-omega and theta (1.5 hours)
- Techniques for estimating, calculating and empirically determining algorithm complexity (2 hours)
- Recursive algorithms: design, analysis and correctness proofs (2.5 hours)
- Techniques of solving recurrence relations (2 hours)
- Algorithms for sorting: – insertion, bubble, selection, quick, merge, heap, counting, radix and bucket (4 hours)
- Data structures and algorithms for searching – binary search tree, B-tree (2 hours)
• Algorithms for optimization (dynamic programming): factory scheduling, matrix multiplication, longest common subsequence (3 hours)
• Algorithms for optimization (greedy algorithms): activity scheduling, knapsack, Huffman coding (2.5 hours)
• Priority queue data structures and algorithms: binary heaps, binomial heaps (2.5 hours)
• Disjoint set data structure and union/find algorithms (1.5 hours)
• Graph representations and algorithms – adjacency list, adjacency matrix, breadth-first search, depth-first search, topological sort (2.5 hours)
• Minimum spanning trees – Prim's and Kruskal's algorithms (1 hour)
• Conclusion: complexity levels, P, NP and NP-completeness, undecidable problems (3 hours)
• Exams (4 hours)

Course Requirements
• Programming Assignments (30%)
• Homework Assignments (30%)
• Exams (40%)

Syllabus prepared: Spring 2016