Computer Science Department
cs.salemstate.edu

CSC 715 Analysis of Algorithms **3 cr.**

Catalog Description:

This course presents a variety of general algorithms in the computing field, examines the design and implementation techniques of useful and efficient algorithms, and analyzes algorithmic complexity. Topics include mathematical tools for algorithm analysis, numeric algorithms, tree structures, hashing techniques and recursion, analysis of searching and sorting algorithms, dynamic programming, graph representation and traversal algorithms, pattern matching, computation complexity, and computational geometry. Three lecture hours per week.

Course Prerequisites: graduate status or permission of graduate program coordinator.

Course Goals:

The aims of this course are:

CG01: to present a coherent view of advanced concepts of data structures and general algorithms;
CG02: to assess the efficiency and complexity of algorithms;
CG03: to implement algorithms in one of the programming languages.

Course Objectives:

Upon completion of the course, the student will have demonstrated the ability to:

CO01: perform best case, worst case, and average case analysis of selected algorithms for managing lists, trees, heaps, and hash tables;
CO02: implement and use greedy algorithms and dynamic programming techniques;
CO03: explain and use the techniques of probabilistic and amortized analysis;
CO04: explain and use different searching and sorting algorithms, minimum spanning trees, and shortest-path algorithms;
CO05: understand and use in practical projects the techniques presented in string matching, computational geometry, and graph representation and traversals;
CO06: implement and apply several selected algorithms in real world problems.

Topics Agenda:

The course topics will be covered in the class in one semester (fifteen weeks) as follows:

Week01: Review of important topics covered in undergraduate courses Data Structure and Algorithms and Discrete Structures (particularly mathematical induction).
Week02: Introduction of mathematical tools for algorithms and analysis such as big-O notation, summation formulas, recursion relations and functions.
Week03: Numeric algorithms (polynomial arithmetic, matrix operations, random numbers, Fast Fourier Transform, etc).
Week04: Analysis of searching and sorting algorithms (linear search, binary search, and different sort algorithms).
Week05: Tree structures and their corresponding algorithms (balancing algorithms, traversal algorithms).
Week06: Hashing techniques and recursions.
Week07: Probabilistic analysis; reviews and midterm exam.
Week08: Graph representations and elementary graph algorithms (breadth-first search, depth-first search).
Week09: Graph traversal algorithms (minimum spanning trees, shortest-path algorithms).
Week10: Pattern (string) matching algorithms.
Week11: Greedy methods, amortized analysis, dynamic programming.
Week12: Geometric algorithms (convex hull, closest points, intersection problems, etc.).
Week13: Computational complexity, NP-completeness.
Week14: Applications of some of selected algorithms.
Week15: Presentations, reviews, final exam.

Testing and Grading:

Student's performance in the course and understanding of the course topics will be evaluated through different types of measurements: written assignments, programming assignments (projects), midterm and final examinations. Written assignments will consist of short answer questions and brief essay assignments to be used to evaluate the students' understanding of important course topics and ability of finding existing solutions to the algorithmic problems in the related areas. Programming assignments will test student's ability in application of the algorithms in real world problems. Two comprehensive examinations, midterm and final exam, will be conducted to evaluate students' understanding of the concepts of algorithms and problem-solving skills in the application of the algorithms and solutions covered in this course.

The final grade will be determined using the following weights:

- Written Homework 30%
- Programming Assignments 30%
- Midterm Exam 15%
- Final Exam 25%

Bibliography:

Cormen, Thomas H. **Algorithms Unclocked.** The MIT Press, 2013

Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford. **Introduction to Algorithms**. **Third Edition.** The MIT Press, 2010.

Edmonds, Jeff. **How to Think About Algorithms.** Cambridge University Press, 2008.

Goodrich, Michael T. **Algorithm Design and Applications. First Edition.** Wiley, 2014.

Greene, Daniel; Knuth, Donald. **Mathematics for the Analysis of Algorithms. Second Edition.** Birkhauser Boston, 2007

Heineman, George T.; Pollice, Gary, Selkow, Stanley. **Algorithms in a Nutshell**. O'Reilly Media, 2008.

Knuth, Donald. **The Art of Computer Programming**, **Volume 1 Third Edition and Volume 3 Second Edition**. Addison-Wesley,1997-98.

McConnell, Jeffrey. **Analysis of Algorithms.** Second Edition. Jones & Bartlett, 2008.

Moret, B. M. E.; Shapiro, H. D. **Algorithms from P to NP**, Volume I: Design and Efficiency. Benjamin/Cummings, 1991.

Neapolitan, Richard. **Foundations of Algorithms. Fifth Edition**. Jones & Barlett Learning, 2014.

Purdom, Paul Walton, Jr.. **The Analysis of Algorithms** Oxford University Press, 2004.

Rawlins, Gregory J. E. **Compared to What? An Introduction to the Analysis of Algorithms.** Computer Science Press W. H. Freeman, 1992.

Sedgewick, Robert. **An Introduction to the Analysis of Algorithms. Second Edition.** Addison-Wesley, 2013.

Sedgewick, Robert. **Algorithms. Fourth Edition.** Addison-Wesley, 2011.
Part 5. Third Edition. Addison-Wesley, 2002.

Sedgewick, Robert. **Algorithms in Java, Parts 1-4. Third Edition.** Addison-Wesley, 2001.
Part 5. Third Edition. Addison-Wesley, 2002.

Sedgewick, Robert. **Algorithms in Java, Parts 5: Graph Algorithms Third Edition.** Addison-Wesley, 2003.
Part 5. Third Edition. Addison-Wesley, 2002.

Skiena, Steven S.; Revilla, Miguel. **The Algorithm Design Manual. Second Edition**. Springer, 2010.

Skiena, Steven S.; Revilla, Miguel. **Programming Challenges**. Springer, 2003.

Stephens, Rod. **Essential Algorithms: a Practical Approach to Computer Algorithms. First Edition**. Wiley, 2013.