Computer Science Department
cs.salemstate.edu

CSC 415 Analysis of Algorithms **3 cr.**

Instructor: TBA **Office:** location **Phone:** (978) 542-extension
email: TBA@salemstate.edu **Office Hours:** days and times

Section	Time	Room	Final Exam
nn	days and times	location	date and time

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 plus programming work outside the class.

Prerequisites: CSC 260 and MAT 214A.

Goals:

The aims of this course are:

CG01: to present a coherent view of algorithms encountered in earlier courses;
CG02: to introduce a selection of more advanced algorithms;
CG03: to present common tools and techniques for assessing the costs and complexity of algorithms.

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;
CO02: explain and analyze examples of greedy algorithms;
CO03: explain and use the main ideas of probabilistic and amortized analysis;
CO04: explain the analysis and use of breadth-first and depth-first search, minimum spanning trees, and shortest-path algorithms;
CO05: understand and use introduced algorithms in such applications as string matching, computational geometry, dynamic programming, and graphs.

Student Outcome (SO) vs. Course Objectives matrix

SO	CO01	CO02	CO03	CO04	CO05
SO-1	✓	✓	✓	✓	✓
SO-2	✓	✓	✓	✓	✓
SO-3					
SO-4					
SO-5					
SO-6	✓	✓	✓	✓	✓

Notes:

SO-1: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
SO-2: Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
SO-3: Communicate effectively in a variety of professional contexts.

SO-4: Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
SO-5: Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
SO-6: Apply computer science theory and software development fundamentals to produce computing-based solutions.

Topics:

• brief review of course programming language (as appropriate)	
• mathematical tools for algorithm analysis	**DS3(3), DS4(2)**
° limits and big-O notation	
° summation formulas	
° recursion relations	
° generating functions	
• finite automata	**AL7(6)**
° deterministic (DFA)	
° non-deterministic (NDFA)	
° non-deterministic with empty moves (NDFAe)	
• turing machines	
• numeric algorithms (Fibonacci numbers, polynomial arithmetic, Fast Fourier Transform, random numbers etc.)	**AL3(16)**
• analysis of searching and sorting algorithms	
° finding largest (or smallest), second largest, etc.	
° linear search, binary search	
° insertion sort, quicksort, merge sort, tree sort, heap sort	
• multi-lists	
• tree structures	
° general trees	
° *n*-way trees	
° balancing algorithms	
° traversal algorithms	
° expression trees	
• review of basic hashing techniques	
• advanced fast-access algorithms	
• graphs, their representations, and traversal algorithms	**AL3(4)**
° graph representation	
° elementary graph algorithms	
° minimum spanning trees	
° shortest-path algorithms	
• pattern matching and string algorithms	**AL2(3)**
• dynamic programming	AL8(2)
• geometric algorithms	AL8(2)
• computational complexity, NP-completeness	AL6(2)

Assignments and Examination:

The primary emphasis of this course is on the understanding of a set of algorithms with wide-ranging applicability, together with their costs, advantages, and disadvantages. Each algorithm is presented and discussed in detail, including classic analysis to determine best, average, and worst-case performance. Implementation considerations will be touched on, along with proper solution development. Algorithms may be assigned for implementation in programming projects which will include analysis of time and space costs. (Possible programming assignments: creation and evaluation of expression trees; balancing algorithms; breadth-first vs. depth-first graph traversal; hashing, graph algorithms, pattern matching algorithms, and geometric algorithms.) There will be periodic written homework assignments on the major topics of the course.

The course grade will be determined using the following approximate weights: programming projects- 40%, written homework - 30%, final examination - 30%

Course Objective / Assessment Mechanism matrix

	Programming Projects	Written Homework	Final Examination
CO01	✓	✓	✓
CO02		✓	✓
CO03		✓	
CO04	✓	✓	✓
CO05	✓	✓	✓

Bibliography:

[Mathematical background]

Graham, Ronald; Knuth, Donald; Patashnik, Oren. **Concrete Mathematics: a Foundation for Computer Science. Second Edition.** Addison-Wesley Professional, 1994.

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

[Analysis of algorithms]

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.

Heineman, George T.; Pollice, Gary, Selkow, Stanley. **Algorithms in a Nutshell: A Practical Guide. Second Edition** O'Reilly Media, 2016

Levitin, Anany. **Introduction to the Design and Analysis of Algorithms. Third Edition.** Pearson 2011.

Kronsjo, Lydia. **Algorithms: Their Complexity and Efficiency**. **Second Edition.** John Wiley, 1987.

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

Miller, Russ; Boxer, Laurence. **Algorithms Sequential & Parallel: A Unified Approach. Third Edition.** Cengage Learning, 2012

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

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

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.

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

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

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

Weiss A. Mark. **Data Structures and Algorithm Analysis in Java. Third Edition.** Pearson 2012.

Academic Integrity Statement:

"Salem State University assumes that all students come to the University with serious educational intent and expects them to be mature, responsible individuals who will exhibit high standards of honesty and personal conduct in their academic life. All forms of academic dishonesty are considered to be serious offences against the University community. The University will apply sanctions when student conduct interferes with the University primary responsibility of ensuring its educational objectives." Consult the University catalog for further details on Academic Integrity Regulations and, in particular, the University definition of academic dishonesty.

The Academic Integrity Policy and Regulations can be found in the University Catalog and on the University website (http://catalog.salemstate.edu/content.php?catoid=13&navoid=1295#Academic_Integrity). The formal regulations are extensive and detailed - familiarize yourself with them if you have not previously done so. A concise summary of and direct quote from the regulations: "Materials (written or otherwise) submitted to fulfill academic requirements must represent a student's own efforts". *Submission of other's work as one's own without proper attribution is in direct violation of the University's Policy* and

will be dealt with according to the University's formal Procedures. *Copying without attribution is considered cheating in an academic environment - simply put,* ***do not do it!***

University-Declared Critical Emergency Statement:

In the event of a university-declared emergency, Salem State University reserves the right to alter this course plan. Students should refer to www.salemstate.edu for further information and updates. The course attendance policy stays in effect until there is a university-declared critical emergency.

In the event of an emergency, please refer to the alternative educational plans for this course, which will be distributed via standing class communication protocols. Students should review the plans and act accordingly. Any required material that may be necessary will have been previously distributed to students electronically or will be made available as needed via email and/or Internet access.

Equal Access Statement:

"Salem State University is committed to providing equal access to the educational experience for all students in compliance with Section 504 of The Rehabilitation Act and The Americans with Disabilities Act and to providing all reasonable academic accommodations, aids and adjustments. **Any student who has a documented disability requiring an accommodation, aid or adjustment should speak with the instructor immediately.** Students with Disabilities who have not previously done so should provide documentation to and schedule an appointment with the Office for Students with Disabilities and obtain appropriate services."

Note: This syllabus represents the intended structure of the course for the semester. If changes are necessary, students will be notified in writing and via email.