UNIVERSITY OF NEBRASKA AT OMAHA
COURSE SYLLABUS/DESCRIPTION

Department and Course Number	CSCI 4100
Course Title	Introduction to Algorithms
Course Coordinator	Parvathi Chundi
Total Credits	3
Date of Last Revision	Mar 5th, 2014

1.0 Course Description

1.1 Catalog description.
The course provides students a basic understanding of algorithm analyses. Main topics include: growth of functions, asymptotic notation, recurrences, divide and conquer, sorting and its lower bounds, dynamic programming, greedy algorithms, and graph traversal.

1.2 Prerequisites of the course (Courses).
CSCI 3320
Fundamentals of logic
Fundamental proof techniques
Functions and Relations
Arrays, lists, trees, graphs, stacks, queues
Summations and Series
Counting techniques

1.3 Overview of content and purpose of the course.
In this course, a comprehensive introduction to the study of computer algorithms will be presented. Mathematical tools needed to analyze the efficiency of a computer algorithm will be discussed including logarithms, series and sequences, recurrences, and induction. Different types of algorithms will be covered in considerable depth so that students are able to analyze an algorithm for its time/space complexity as well as correctness. Students will study sorting and searching algorithms, algorithmic techniques such as dynamic programming and greedy techniques, number theoretic algorithms such as the GCD, geometry algorithms such as finding the convex hull. The course will also cover algorithms for graph traversal and decomposition into connected components.

1.4 Unusual circumstances of the course.
None.

2.0 Course Justification Information

2.1 Anticipated audience / demand.
The course is intended for junior and senior level undergraduate students in computer science (CS). The course is also for junior and senior level students in other disciplines so long as they have the basic knowledge of data structures and discrete mathematics.

2.2 Indicate how often this course will be offered and the anticipated enrollment.
Enrollment: 25
Course rotation: once a year

2.3 If it is a significant change to an existing course, please explain why it is needed.
This is a new course.

3.0 Objectives

List of performance objectives stated in terms of the student educational outcomes.

3.1 Be able to analyze an algorithm to compute its time complexity.
3.2 Be able to analyze an algorithm to compute its space complexity.
3.3 Be able to state and prove the correctness of algorithms.
3.4 Understand the need to design efficient algorithms.
3.5 Understand the tradeoffs among sorting algorithms.
3.6 Be able to construct solutions to problems using dynamic programming and greedy techniques.
3.7 Be able to apply graph and number theoretic algorithms to solve problems in diverse applications.

4.0 Content and Organization

4.1 Introduction ` (1.5 hours)
- 4.1.1 Algorithms as Technology
- 4.1.2 Designing Algorithms

4.2 Analysis Techniques (9 hours)
- 4.2.1 RAM Model of Computation
- 4.2.2 Analysis of Sequential programs
- 4.2.3 Asymptotic Notation
- 4.2.4 Standard Notations and Common Functions
- 4.2.5 Analysis of Recursive Programs
- 4.2.6 Substitution method
- 4.2.7 Recursion Tree Method
- 4.2.8 Master Method

4.3 Sorting and Searching (3 hours)
- 4.3.1 Heap Sort and Analysis
- 4.3.2 Quick Sort and Analysis
- 4.3.3 Sorting in Linear Time
- 4.3.4 Lower Bounds on Sorting
- 4.3.5 Medians and Order Statistics

4.4 Dynamic Programming (9 hours)
- 4.4.1 Sub-problems and optimal substructure
- 4.4.2 Matrix Chain Multiplication Problem
- 4.4.3 Longest Common Subsequence Problem

4.5 Greedy Algorithms (4.5 hours)
- 4.5.1 Activity Selection Problem
- 4.5.2 Huffman codes

4.6 Number Theoretic Algorithms (9.0 hours)
  4.6.1 Greatest Common Divisor
  4.6.2 Modular Arithmetic
  4.6.3 RSA Public Cryptographic System
  4.6.4 The Chinese Remainder Theorem
4.7 Computational Geometry (4.5 hours)
  4.7.1 Intersecting line segments
  4.7.2 Finding Convex hull
  4.7.3 Finding closest Pair of Points
4.8 Elementary Graph Algorithms (7.5 hours)
  4.8.1 Representation of graphs
  4.8.2 Breadth-first traversal
  4.8.3 Depth-first traversal
  4.8.4 Topological sort
  4.8.5 Strongly connected components.

5.0 Teaching Methodology

5.1 The basic teaching method will be the instructor's lectures.

5.2 Student role in the course.
The students will attend lectures, participate in classroom activities, complete assignments and term projects, and pass all exams.

5.3 Contact hours.
Three (3) hours a week.

6.0 Evaluation

Students are evaluated based on their understanding of materials covered in the semester. Basis of evaluation includes understanding of basic concepts analyses methods, growth of functions, and design of efficient algorithms involving the basic algorithms and techniques discussed in class including sorting, order statistics, dynamic programming, greedy algorithms, computational geometry, and number theoretic methods.

6.1 Basis for determining the final grade (Course requirements and grading standards) specifying distinction between undergraduate and graduate, if applicable.
The weights of grade will be allocated for the categories of participation, homework, projects and exams.

Exams:	15%
Classroom participation/Home works/quizzes	55%
Final exam:	30 %

6.2 Grading scale and criteria.

97%-100%	A+
94%-96%	A
90%-93%	A-
87%-89%	B+

84%-86%	B
80%-83%	B-
77%-79%	C+
74%-76%	C
70%-73%	C-
67%-69%	D+
64%-66%	D
60%-63%	D-
Below 60%	F

7.0 Resource Material

7.1 Textbooks and/or other required readings used in course.

7.1.1 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms*, McGraw-Hill, 2011.

7.1.2 Robert Sedgewick and Philippe Flajolet, *An Introduction to the Analysis of Algorithms (2nd Edition)*, Pearson Education, 2013.

7.1.3 Jon Klienberg and Eva Tardos, *Algorithm Design*, Addison-Wesley, 2005.

7.2 Other suggested reading materials, if any.

7.3 Other sources of information.
Course Website on Blackboard.

7.4 Current bibliography of resource for student's information.

7.4.1 Steven S. Skiena, The Algorithm Design Manual, Springer, 2008.

7.4.2 Alfred Aho, John Ullman and Jeffery Hopecraft, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

7.4.3 Ellis Horowitz and Sartaj Sahni, *Fundamentals of Computer Algorithms*, Computer Science Press. 1984.

7.4.4 Kurt Mehlhorn and Peter Sanders, *Algorithms and Data Structures: The Basic Toolbox*, Springer 2008.

7.4.5 Richard Neapolitan and Kumarss Naimipour, *Foundations of Algorithms*, Jones & Bartlett Learning, 2010.

7.4.6 Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, *Algorithms*, McGraw-Hill Education, 2008.

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

7.4.8 Udi Manber, Introduction to Algorithms: A creative Approach, Addison-Wesley, 1989.

7.4.9 Robert Sedgewick and Kevin Wayne, *Algorithms, 4th Edition*, Pearson, 2013.

7.4.10 Michael R. Garey and David S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Co. 1979.

7.4.11 Donald E. Knuth, The Art of Computer Programming (Volumes 1-4), Addison-Wesley, 2005.

8.0 **Computer Science Accreditation Board (CSAB) Category Content (class time in hours)**

CSAB Category	*Core*	*Advanced*
Data structures		15.0
Algorithms and software design		30.0

9.0 Oral and Written Communications

Every student is required to submit at least __0__ written reports (not including exams, tests, quizzes, or commented programs) to typically ___0_ pages and to make __0__ oral presentations of typically _0__ minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

10.0 Social and Ethical Issues

Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g. test questions, essays, oral presentations, and so forth?).

No coverage.

11.0 Theoretical content

Please list the types of theoretical material covered, and estimate the time devoted to such coverage.

Proof techniques: 20 contact hours.
Analysis Techniques: 20 contact

12.0 Problem analysis

There will be analysis on quizzes, written assignments and exams. Analysis may be given during the lecturing or in the written form to be distributed in the class. Students are asked to review the analysis materials, and are responsible for these materials.

13.0 Solution design

Students will gain database design experiences using entity-relationship approach and normalization theory through classroom quizzes, written assignments and exams.

CHANGE HISTORY

Date	***Change***	***By whom***	***Comments***
01/08/2014	Initial ABET version	Chundi	
02/19/2014	Updated Reference List.	Chundi	
03/05/2014	Updated course description.	Chundi	