1785

The University of Georgia

Department of Computer Science

Course Information Sheet
CSCI 4470
Algorithms

Brief Course Description (50-words or less)	Algorithms, covering basic analysis techniques, basic design techniques (divide-and-conquer, dynamic programming, greedy), basic and advanced graph algorithms, and NP-completeness theory.
Extended Course Description / Comments Use this section to put additional information that's relevant to whom this course is targeting	
Pre-Requisites and/or Co-Requisites	CSCI 2720 Data Structures And CSCI 2670 Introduction to Theory of Computation
Approved Textbooks (If more than one, course text used during a semester is at the discretion of the instructor)	Author(s): Cormen, Leiserson, Rivest, and Stein Title: Introduction to Algorithms Edition: 3rd ISBN-13: 9780262033848
Specific Learning Outcomes (Performance Indicators)	This course provides an introduction to the modern study of computer algorithms most relevant to students studying computer engineering. At the end of the semester, all students will be able to do the following: 1. Analyze time complexity for algorithms using the asymptotic notations. 2. Analyze sorting and selection algorithms and prove O(nlog n) lower bound for sorting problem on comparison-based model. 3. Design and analyze divide-and-conquer algorithms. 4. Design and prove the optimality of greedy algorithms. 5. Design dynamic programming algorithms, including solution recurrence formulation, iterative, bottom-up design, and development of trace-back subroutines. 6. Apply basic graph search algorithms (depth-first search, breadth first search) to solve basic graph-theoretic problems. 7. Analyze and prove graph algorithms for Minimum Spanning Tree and Shortest Path problems. 8. Prove properties for flow networks and the equivalence between Maximum Flow and Min Cut problems. 9. Analyze and prove for Maximum Flow algorithms. 10. Prove basic properties for the NP-completeness theory. 11. Prove NP-completeness for some combinatorial problems using reduction techniques.

Relationship Between Student Outcomes and Learning Outcomes

		Student Outcomes										
		a	b	c	d	e	f	g	h	i	j	k
Learning Outcomes	1	•	•	•								
	2	•	•	•								
	3	•	•	•						•	•	
	4	•	•	•						•	•	
	5	•	•	•						•	•	
	6	•	•	•						•	•	
	7	•	•	•						•	•	
	8	•	•	•						•	•	
	9	•	•	•						•	•	
	10	•	•	•								
	11	•	•	•								

Program Outcomes

(These are ABET-specified and should not be changed)

a. An ability to apply knowledge of computing and mathematics appropriate to the discipline.
b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
c. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
d. An ability to function effectively on teams to accomplish a common goal.
e. An understanding of professional, ethical, legal, security and social issues and responsibilities.
f. An ability to communicate effectively with a range of audiences.
g. An ability to analyze the local and global impact of computing on individuals, organizations, and society.
h. Recognition of the need for and an ability to engage in continuing professional development.
i. An ability to use current techniques, skills, and tools necessary for computing practice.
j. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
k. An ability to apply design and development principles in the construction of software systems of varying complexity.

Major Topics Covered (Approximate Course Hours) 3 credit hours = 37.5 contact hours 4 credit hours = 50 contact hours Note: Exams count as a major topic covered	Asymptotic notations (4-hours) Sorting and selection algorithms (6-hours) Greedy algorithms (4-hours) Divide-and-Conquer algorithms (4-hours) Dynamic programming algorithms (6-hours) Graph depth-first and breadth-first searches (4-hours) Minimum Spanning Tree algorithms (4-hours) Shortest Paths algorithms (5-hours) Flow networks and Maximum Flow algorithms (6-hours) NP-completeness theory (3-hours) Polynomial time reduction and proofs (4-hours)
Assessment Plan for this Course	Each time this course is offered, the class is initially informed of the Course Outcomes listed in this document, and they are included in the syllabus. At the end of the semester, an anonymous survey is administered to the class where each student is asked to rate how well the outcome was achieved. The choices provided use a 5-point Likert scale containing the following options: Strongly agree, Agree, Neither agree or disagree, disagree, and strongly disagree. The results of the anonymous survey are tabulated and results returned to the instructor of the course. The course instructor takes the results of the survey, combined with sample student responses to homework and final exam questions corresponding to course outcomes, and reports these results to the ABET committee. If necessary, the instructor also writes a recommendation to the ABET committee for better achieving the course outcomes the next time the course is offered.
How Data is Used to Assess Program Outcomes	Each course Learning Outcome, listed above, directly supports one or more of the Student Outcomes, as is listed in "Relationships between Learning Outcomes and Student Outcomes". For CSCI 4470, Student Outcomes (a), (i), and (j) are supported.
Course Master	Dr. Liming Cai
Course History	