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	N/A
Pre-Requisites and/or Co-Requisites	1) CSCI 2720 (Data Structures) OR CSCI 2725 (Data Structures for Data Science) AND 2) CSCI 2670: Introduction to Theory of Computation
Required, Elective or Selected Elective	Selected Elective Course
Approved Textbooks (if more than one listed, the textbook used is up to the instructor's discretion)	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	□	●	●	●								
	□	●	●	●								
	□	●	●	●						●	●	
	□	●	●	●						●	●	
	□	●	●	●						●	●	
	6	●	●	●						●	●	
	7	●	●	●						●	●	
	8	●	●	●						●	●	
	9	●	●	●						●	●	
	10	●	●	●								
	11	●	●	●								

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)

Course Master Dr. Liming Cai