CS 4521: Algorithms and Data Structures (4)

Catalog Description:
Asymptotic analysis of algorithms. Methods for proving correctness. Implementation of algorithms. Survey of algorithms and data structures, such as: heaps and heapsort, quicksort, binary search trees, red-black trees, B-trees, hash tables, graph algorithms, dynamic programming, and greedy algorithms.

Textbook: Thomas Cormen, Charles Leiserson, Ronald Rivest and Clifford Stein, *Introduction to Algorithms*, 2nd Ed., MIT Press, 2001.

Course Goals:
Advanced computer programming requires an understanding of both the objects on which programs work, that is, data structures, and the sequences of steps driving the programs themselves, that is, algorithms. In order for data structures and algorithms to be generally useful, they often must overcome memory and speed limitations of the underlying machine. Thus it is important for the programmer to be able to analyze algorithms for their efficiency in execution and use of space for data. This course will approach the study of data structures and algorithms with an eye toward both analyzing efficiency and developing working models using sound data abstraction and procedural abstraction practices.

Prerequisites by Course & Topic
CS 2511: Software Analysis and Design.- implementation experience, data abstraction, algorithm analysis (implicit requirement, course required by CS 3512)
CS 3512: Computer Science Theory - proof techniques, recursion and mathematical induction, recursive algorithms, analysis of algorithms, assertions and loop invariants, complexity measures of algorithms, combinatorial counting techniques

Major Topics Covered in the Course
- Asymptotic Analysis and Correctness Proofs
- Algorithms and Data Structures

Class/Laboratory Schedule: Lecture: 3 hours per week, Laboratory: 1

Course Outcomes
1. Further develop ability to understand, apply and prove precise mathematical claims about space and time requirements of algorithms, and about correctness of algorithms.
   a. Solidify understanding of definitions and notation for asymptotic notation, including the ability to prove claims involving asymptotic notation..
   b. Improve intuitions about the practical significance of precise claims involving asymptotic notation.
   c. Solidify ability to understand proofs of algorithm correctness, including the use of loop invariant arguments (as a variation on the theme of inductive proofs).
2. Detailed knowledge of a number of standard algorithms, and data structures, including knowledge of time and space complexity, ability to prove (or understand proofs) of properties of data structures and associated algorithms, and also the ability to implement such algorithms.
   a. In-depth knowledge of several of the standard sorting algorithms and associated data structures.
   b. Knowledge of a number of more advanced algorithms and data structures, such as red-black trees, B-trees, binomial heaps, Fibonacci heaps, and graphs.

Relationship to Program Outcomes

CS 4521, an elective course, requires computer science theory as prerequisites. This course contributes to meeting the following program outcomes:

1. *Students understand the mathematics and statistics that underlie scientific applications.*
   Students comprehend the definitions used in standard asymptotic notation, and also comprehend the use of this notation to express time complexity properties of algorithms. Students comprehend and in some cases prove mathematical properties of data structures. Students understand proofs of correctness for some algorithms, including the use of loop invariants. Course outcome 1 maps to this program outcome.

2. *Students can design, develop and analyze significant software systems.*
   Students gain additional experience in software design and implementation by implementing various algorithms and data structures. Ability of students to design, develop and analyze significant software systems is enhanced by knowledge of algorithms and data structures, including their space and time complexity properties. Both course outcomes map to this program outcome.
3. *Students understand the fundamentals of computer organization and architecture, data structures and related algorithms, and programming languages.*
   This course substantially increases a student's knowledge of data structures and algorithms. Both course outcomes map to this program outcome.
4. *Students can apply computer science principles and practices to a variety of problems.*
   Students work with a variety of new data structures and algorithms. They also gain a deeper understanding of asymptotic notation, and of proofs of correctness, including the use of loop invariants, and see a range of applications of these concepts. Both course outcomes map to this program outcome.

Assessment Plan for Course:
This course is assessed every third year by the instructor and a course assessment document covering all of the course outcomes and their effect on the program outcomes is prepared.

Estimate CSAB Category Content

	CORE	ADVANCED		CORE	ADVANCED
Data Structures		2	Computer Organization and Architecture		
Algorithms		2	Concept of Programming Languages		
Software Design					

Coordinator/Prepared by: D. Dunham