CSC 325 Algorithms & Advanced Data Structures

Catalog Description:

This course should be taken as soon as possible after CSC 232. Algorithms and advanced data structures, including graphs, heaps, self-adjusting data structures, set representations and dynamic programming. Sample applications, including memory management and data compression. Introduction to NP-complete problems. Correctness proofs and efficiency analysis are stressed. 3(3-0) F, S

Prerequisites: CSC232: Data Structure
MTH215: Discrete Mathematics or MTH315: Algebraic Structures

This course is a prerequisite for:
CSC 450: Software Engineering
CSC 460: Theory of Computer Operating Systems

Required Texts:

Introduction to Algorithms, Third Edition, by T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, The MIT Press and McGraw-Hill Book Company, 2009.

Major Topics:

1. Algorithm analysis
    a. Asymptotic notations: ω, Ω, θ, O, o
    b. Growth of functions
    c. Solving recurrences
    d. Amortized analysis
2. Algorithm design approaches
    a. Incremental approach
    b. Divide-and-conquer
    c. Randomized algorithms
    d. Dynamic programming
    e. Greedy algorithms
    f. Approximation algorithms
3. Advanced data structures
    a. Hash tables
    b. Priority queues
    c. Balanced binary trees
    d. Disjoint sets
    e. Graphs
4. NP-Completeness

Student Outcomes Assessed in CSC 325

A. Students will attain an ability to apply knowledge of computing and mathematics appropriate to the discipline.
B. Students will attain an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
C. Students will attain an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
I. Students will attain an ability to use current techniques, skills, and tools necessary for computing practice.
J. Students will attain 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

Table 1. Student Outcomes Assessed in CSC 325

CSC 325 Student Outcomes	CSC 325 Performance Indicators	CSC 325 Assessment Goals
CSC 325 contributes to SO A: Students will attain an ability to apply knowledge of computing and mathematics appropriate to the discipline	**PI 325-1-a:** recognize the behavior of functions (e.g. identify that if a function is monotonically increasing or not)	**PI 325-1-a:** 80% >= pass
	PI 325-1-b: understand asymptotic notations (e.g. identify that if a function belongs to the set of the little-o of another function)	**PI 325-1-b:** 60% >= pass
	PI 325-1-c: analyze the time complexity of a given algorithm	**PI 325-1-c:** 60% >= pass
CSC 325 contributes to SO B: Students will attain an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution	**PI 325-2-a:** design an algorithm to solve the 1^{st} order statistic problem	**PI 325-2-a:** 80% >= pass
	PI 325-2-b: design an algorithm to solve the n^{th} order statistic problem	**PI 325-2-b:** 70% >= pass
CSC 325 contributes to SO C: Students will attain an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs	**PI 325-3:** write a program to find all connected components in a graph using disjoint-set operations with rank heuristic and path compression.	**PI 325-3:** 80% >= pass
CSC 325 contributes to SO I: Students will attain an ability to use current techniques, skills, and tools necessary for computing practice	**PI 325-4:** use dynamic programming method to find a longest common subsequence	**PI 325-4:** 70% >= pass
CSC 325 contributes to SO J: Students will attain 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	**PI 325-5:** recognize in-place *vs.* non in-place sorting algorithms	**PI 325-5:** 70% >= pass (no more than two errors)