0

Department of Computer Engineering

Algorithms &Computational Complexity (66314)

Total Credits	3

major elective

Prerequisites	P1 : Data Structure and Algorithms (66211)

Course Contents

Introduction to Algorithms, Algorithms as a Technology, Components of an Algorithm, Algorithms Efficiency, Insertion Sort, Merge Sort. Growth of Functions, Recurrences, Sorting and Order Statistics, Data Structures, Advanced Design and Analysis Techniques(Dynamic Programming, Greedy Algorithms), Graph Algorithms and NP-Completeness

	Intended Learning Outcomes (ILO's)	Student Outcomes (SO's)	Contribution
1	The ability to apply knowledge of basic terms associated with algorithms, such as Growth of Functions, Recurrences, Big-O, Big-Omega, Big-Theta, etc.	A	20 %
2	The ability to understand, design and analyze various sorting algorithms such as: insertion sort, merge sort, quick sort, heap sort, etc, in addition to various searching algorithms such as binary, breadth-first, depth-first searches.	C	30 %
3	The ability to understand, design and analyze various basic data structures (stack, queue, tree, etc.) and advanceddata structures (Hash Tables, BSTrees, Red-Black and Skip Lists).	C	20 %
4	The ability to understand and utilize Advanced Design and Analysis Techniques (Greedy Algorithms, Dynamic programming, Graph algorithms, Amortized Analysis and Linear Programming) for solving practical problems.	K	30 %

Textbook and/ or Refrences

Introduction to Algorithms, Third Edition, MIT Press 2010

Assessment Criteria	Percent (%)
Mid. Term Exam	45 %
Homeworks	15 %
Final Exam	40 %

Course Plan

Week	Topic
1	Introduction - Insertion Sort; Analyzing and Designing Algorithms - Merge Sort (Divide and Conquer).
2	Growth of Functions&Asymptotic Notations: - Complexity (Worst-Case, Best-Case, and Average-Case), Growth Rates of Functions (Polynomials, Exponents, Logarithms, Summation, and Factorials). - Asymptotic Notations (Big-O, Omega and Theta).
3-4	Solving Recurrences - Recurrence Trees - Substitution Method - Master Method.
5-6	Divide and Conquer Techniques: - Binary Search, Powering a number, Fibonacci, Matrix Multiplication - Quick Sort
7	Heap Sort (Heap Property; Build Heap; Heap Sort; Applications: Max-Priority Queues; Basic Operations on Max-Priority Queues (Max, Extract-Max, Increase-Key, and Insert)).
8	Linear Sorting: Counting Sort &RadixSort
8	First Exam

9	Review of Basic Data Structures: Stack, Queue, Tree. Single and Multiple Array Representations.
10-11	Advanced Data Structures: Hash Tables, BSTrees, Red-Black and Skip Lists
12-14	Dynamic Programming&GreedyalgorithmsAdvanced Design and Analysis Techniques
14	Second Exam
15	Graphs &MST
16	Amortized Analysis, Linear Programming&NPCompleteness
16	Final Exam