Faculty of Computers and Information Technology

Algorithms

Information :

Course Code :	CSC 313	**Level :**	Undergraduate	**Course Hours :**	3.00- Hours

Department : Department of Computer Science

Instructor Information :

Title	Name	Office hours
Associate Professor	Wael Hassan gomaa Mohamed Abuzaid	
Assistant Lecturer	YASMIN AMR AHMED ANWAR ALI BADR	5

Area Of Study :

The course is concerned with design and analysis of algorithms. It covers design techniques, Such as dynamic programming and greedy methods, As well as fundamentals of analyzing algorithms for correctness and time and space bounds. Topics include advanced sorting and searching methods, Graph algorithms and geometric algorithms, Notion of an algorithm: Big-o, Small-o, Theta and omega notations. Space and time complexities of an algorithm. Fundamental design paradigms: Divide and conquer Branch and bound, Backtracking dynamic programming greedy methods, Simulation. Theory of up-completeness, Notion of an intractable problem. Measures of approximation: Ratio bound and relative error. Polynomial time approximation scheme. Illustrative examples: Graph theory, Areas vary from year to year, and may include matrix manipulations, String and pattern matching, set algorithms, Polynomial computations, and the fast Fourier transform. Recent correlated software packages should be used through labs.

Description :

The course is concerned with design and analysis of algorithms. It covers design techniques, Such as dynamic programming and greedy methods, As well as fundamentals of analyzing algorithms for correctness and time and space bounds. Topics include advanced sorting and searching methods, Graph algorithms and geometric algorithms, Notion of an algorithm: Big-o, Small-o, Theta and omega notations. Space and time complexities of an algorithm. Fundamental design paradigms: Divide and conquer, Branch and bound, Backtracking dynamic programming greedy methods, Simulation. Theory of up-completeness, Notion of an intractable problem. Measures of approximation: Ratio bound and relative error. Polynomial time approximation scheme. Illustrative examples: Graph theory, Areas vary from year to year, and may include matrix manipulations, String and pattern matching, set algorithms, Polynomial computations, and the fast Fourier transform. Recent correlated software packages should be used through labs

Course outcomes :

a.Knowledge and Understanding: :	
1 -	Measure relative performance of a given algorithm
2 -	Compare and differentiate between different time complexities
3 -	Apply Divide and Conquer technique for problem solving
b.Intellectual Skills: :	
1 -	Up- to- date technology will be applied during the delivery of each course using power point slides for lectures, up -to- date textbooks, and handouts will be provided to disseminate knowledge among students about the different topics of the subject material.

c.Professional and Practical Skills: :	
1 -	Implement more than one Sorting algorithm and enhance the programming skills
2 -	Case studies, work experience, projects, demonstrations, group study, simulations (e.g. computer based), workshops, training, discussions and debate will be implemented through the different courses in order to develop studentsôcapabilities to use ideas and information related to their program of study
d.General and Transferable Skills: :	
1 -	In order to facilitate the personal development of the students, self-assessment through activities such as structured group activities with role play will be used in the different courses
2 -	To develop studentsôabilities to generate ideas and evidence, students will be encouraged to participate in workshops and research projects
3 -	Guest lecturers and guest speakers will be invited to enhance students knowledge about specific subject material as necessary

Course Topic And Contents :

Topic	No. of hours	Lecture	Tutorial / Practical
Introduction to Algorithms and Growth of Functions	3	2	2
Insertion and Selection Sort	3	2	2
Searching algorithms: Linear Search and Binary Search Tree	3	2	2
Merge and Heap Sort	3	2	2
Quick Sort	3	2	2
Mid Term Exam1	2	1	2
Graph Algorithms: Breadth-First Search	3	2	2
Graph Algorithms: Depth-First Search	3	2	2
Graph Algorithms: Minimum Spanning Trees and Shortest Path (Dijkstraôs algorithm)	3	2	2
Geometric Algorithms: Finding the Convex Hull	3	2	2
Geometric Algorithms: Line-Segment Intersections (Map Overlay)	3	2	2
Mid Term Exam2	2	1	2
Geometric Algorithms: Voronoi Diagrams	3	2	2

Teaching And Learning Methodologies :

Lectures
Practical training
Presentation
Exercises
Open Discussion
Projects
Web-Site searches
Case Study

Course Assessment :

Methods of assessment	Relative weight %	Week No	Assess What
Final Exam	40.00	15	
Laboratory Project	20.00	14	
Mid-Term Exam1	15.00	6	
Mid-Term Exam2	15.00	12	
Quizzes	10.00	3	

Recommended books :

Computational Geometry: Algorithms and Applications