Faculty of Computers and Information Technology

Algorithms

Information :

Course Code :	CS313	**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	1
Assistant Lecturer	YASMIN AMR AHMED ANWAR ALI BADR	

Area Of Study :

Apply the basic concepts and theories of algorithms.
Analyze the requirements of a computing system and design algorithms that are appropriate for problems that they might encounter.
Recognize knowledge that enhances skills in fundamental area of computer science.
Use and adopt fundamental and advanced mathematics, basic sciences and algorithms theories in all development phases of computer-based systems.
Solve problems using algorithms through computational analysis and complexities.

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 -	Discuss the fundamental concepts and theories related to algorithms.
2 -	Describe the basic concepts of high level programming languages for algorithms implementation.
3 -	Discuss specifications and strategic planning to solve computational problems, including Divide-and-Conquer, dynamic programming, greedy algorithms.
4 -	Identify the fundamental algorithms required to solve problems in computer science.
b.Intellectual Skills: :	
1 -	Analyze and design algorithms for computing problems considering limitations and constrains.
2 -	Apply and Implement algorithms for CS Problems with in commercial and industrial constrains.
3 -	Propose a set of alternative algorithms for a given problem associated with their results

4 -	Compare and differentiate between algorithms, methods and techniques used in Computer Science Problems solutions.
c.Professional and Practical Skills: :	
1 -	Deploy and develop different algorithms in projects.
2 -	Apply, design methodologies, programming languages, algorithms and different supporting tools.
3 -	Evaluate the quality of computing systems using different factors and different constrains.
d.General and Transferable Skills: :	
1 -	Exploit a range of learning algorithms.
2 -	Utilize effectively general computing facilities.
3 -	Appreciate continuous professional development and lifelong learning.

ABET Course outcomes :

1 -	Demonstrate understanding of design concepts and principles of contemporary computing algorithms.
2 -	Analyze complexity and efficiency of algorithms.
3 -	Analyze the requirements of computing problems to develop their appropriate algorithmic solutions.
4 -	Compare, evaluate and select appropriate algorithms for given computing problems.
5 -	Use and adopt fundamentals and advanced mathematics, basic sciences and algorithm approache theories in the development phases of computer-based systems.
6 -	Solve problems using algorithms through computational analysis and complexities.
7 -	Design different algorithmic approaches; such as sequential, divide-and-conquer, greedy, and dynamic programming

Course Topic And Contents :

Topic	No. of hours	Lecture	Tutorial / Practical
Asymptotic Complexity Analysis	4	2	2
Analysis of traditional Searching and Quadratic Sorting Algorithms	4	2	2
DaC Intro + Merge Sort Algorithm	4	2	2
DaC: Quick/Counting Sort Algorithms	4	2	2
DaC/DP: Fibonacci	4	2	2
DaC/DP: Rod-cutting	4	2	2
DP: Longest Common Subsequence	4	2	2
DP: Matrix Chain Multiplication	4	2	2
Mid-Term Exam	2		
Greedy: Activity Scheduling Problem	4	2	2
Greedy: Knapsack Problem	4	2	2
Branch & Bound Algorithms	4	2	2
Graph Algorithms	4	2	2
Final Exam	2		

Teaching And Learning Methodologies :

Interactive Lectures including discussion

Practical Lab Sessions
Self-Study (Project / Reading Materials / Online Material / Presentations)
Problem Solving

Course Assessment :

Methods of assessment	Relative weight %	Week No	Assess What
Assignments	10.00	3	
Final Exam	40.00	14	
Midterm Exam (s)	30.00	9	
Presentations	5.00	13	
Quizzes	5.00	5	
Team Work Projects	10.00	11	

Course Notes :

An Electronic form of the Course Notes and all the slides of the Lectures is available on the Students Learning Management System (Moodle)

Recommended books :

Richard Neapolitan and Kumarss Naimipour, ȓFoundations of Algorithms Using C++ Pseudocode", Jones & Bartlett Learning, (last edition).

Web Sites :

www.ekb.eg