



# Capital University of Science and Technology

## Department of Computer Science

---

### CS3163 - Design & Analysis of Algorithms

<b>Course Title:</b>	Design & Analysis of Algorithms (CS3163)
<b>Pre-requisite(s):</b>	Data Structures (CS2143)
<b>Credit Hours:</b>	3
<b>Instructor(s):</b>	
<b>Text Book(s):</b>	Introduction to Design & Analysis of Algorithms By Anany Levitin (Third Edition)
<b>Reference Book(s):</b>	<ul style="list-style-type: none"><li>• <b>Foundations of Algorithms</b> By Richard E. Neapolitan, Kumarss Naimipour, Northeastern Illinois University D. C. Heath and Company Lexington, Massachusetts Toronto</li><li>• <b>Data Structures and Algorithm Analysis in C++</b> By Mark Allen Weiss Benjamin/Cummings Publishing Company, Inc.</li></ul>
<b>Web Reference:</b>	<ul style="list-style-type: none"><li>• <a href="https://www.guru99.com/design-analysis-algorithms-tutorial.html">https://www.guru99.com/design-analysis-algorithms-tutorial.html</a></li></ul>

### Course Introduction:

This is one of the cores and very important courses in the science of computing. To become a successful computer scientist/programmer, you need to know major algorithms from different areas of computing. Also, you should be able to design new algorithms and analyze their efficiency (performance). This course is going to equip you with both the needs, i.e., knowledge of the well-known existing algorithms, and train you in designing and analyzing algorithms.

The focus will be on efficient algorithm design. A number of ideas and techniques useful for designing and analyzing algorithms of different well-known problems will be discussed. In particular, the asymptotic time complexity for finding upper and lower bounds of algorithms will be taught. The focus of the course will be on the theoretical foundation for performance analysis of algorithms. Sufficient time will also be spent on the design and implementation of algorithms.



# Capital University of Science and Technology

## Department of Computer Science

---

### Course Objectives:

At the end of this course, the students should be able to apply important design paradigms (Brute Force, Divide & Conquer, Dynamic Programming, Greedy Algorithms, Backtracking, Branch & Bound) for the design of the algorithm. Students can analyze the time and space complexity (performance) of different algorithms by using asymptotic analysis techniques. Students can synthesize efficient algorithms by applying common design and programming techniques.

### Course Learning Outcomes (CLOs):

At the end of this course, the students should be able to:

1. CLO:1 **Apply** important design paradigms (Brute Force, Divide & Conquer, Dynamic Programming, Greedy Algorithms, Backtracking, Branch & Bound) for the design of algorithms [C3-Application]
2. CLO:2 **Analyze** the time and space complexity (performance) of different algorithms by using asymptotic analysis techniques [C4-Analysis]
3. CLO:3 **Synthesize** efficient algorithms by applying common design and programming techniques. [C4- Synthesize]

### CLOs – PLOs Mapping:

	CLO:1	CLO:2	CLO:3
PLO:1 (Academic Education)			
PLO:2 (Knowledge for Solving Computing Problems)	√		
PLO:3 (Problem Analysis)		√	
PLO:4 (Design/Development of Solutions)			√
PLO:5 (Modern Tool Usage)			



# Capital University of Science and Technology

## Department of Computer Science

### Course Contents:

Week	Content
1	<b>Introduction to the course and course objectives</b>
	What is an algorithm?
	Fundamental of the Algorithmic Problem Solving
	Important Problem Types
	Fundamental Data structures
2,3	<b>Fundamentals of the Analysis of Algorithm Efficiency</b>
	The analysis framework
	Asymptotic Notations and Basic Efficiency Classes
	Mathematical Analysis of No recursive Algorithms
	Mathematical Analysis of Recursive Algorithms
	Substitution method
	Iteration method
4,5	<b>Brute Force and Exhaustive Search</b>
	Selection Sort and Bubble Sort
	Sequential Search and Brute-Force String Matching
	Exhaustive Search
	Depth first search
	Breadth first search
6,7	<b>Decrease-and-Conquer</b>
	Insertion Sort
	Topological Sort
	Algorithms for Generating Combinatorial Objects
8	<b>Divide and conquer</b>
	Merge Sort
	Quick Sort



# Capital University of Science and Technology

## Department of Computer Science

	Binary Tree Traversal
	Multiplication of Large Integers
<b>Mid Term Exam</b>	
9,10	<b>Space and Time Trade-Offs</b>
	Sorting by Counting
	Input Enhancement in String Matching
11, 12	<b>Dynamic Programming</b>
	Introduction to the DP Paradigm with example 1
	Coin row problem
	Coin collection problem
	Warshall Algorithm
	Flyod Algorithm
13,14	<b>Greedy Technique</b>
	Prim's Algorithm
	Kruskal's Algorithm
	Dijkstra's Algorithm
15,16	<b>Coping with the Limitations of Algorithm Power</b>
	Backtracking (n-Queen Problem, Hamiltonian Circuit Problem, Subset-Sum Problem)

### Grading Policy:

S.No	Grading	% of Total Marks
1	Assignments	15
2	Quizzes	15
3	Project	10
4	Mid-term Exam	20
5	Final Exam	40
	<b>Total</b>	<b>100</b>