Course Title	**Analysis of Algorithms**		
Course Code	**CC-313**		
Credit Hours	3 (3,0)		
Category	Computing Core		
Prerequisite	CC-213 Data Structures		
Co-Requisite	None		
Follow Up	None		
Course Introduction	Detailed study of the basic notions of the design of algorithms and the underlying data structures. Several measures of complexity are introduced. Emphasis on the structure, complexity, and efficiency of algorithms.		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	**BT**	**PLO**
	CLO1: Explain what is meant by "best", "expected", and "worst" case behavior of an algorithm.	C2 (Explain)	1,2
	CLO2: Identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors.	C3 (Identify)	1,2
	CLO3: Determine informally the time and space complexity of simple algorithms.	C3 (Determine)	2,3
	CLO4: List and contrast standard complexity classes.	C1 (Know)	1.2
	CLO5: Use big O, Omega, Theta notation formally to give asymptotic upper bounds on time and space complexity of algorithms.	C3 (Apply)	3,4
	CLO6: Use of the strategies (brute-force, greedy, divide-and conquer, and dynamic programming) to solve an appropriate problem.	C3 (Apply)	3,4
	CLO7: Solve problems using graph algorithms, including single source and all-pairs shortest paths, and at least one minimum spanning tree algorithm.	C3 (Apply)	3,4
	CLO8: Trace and/or implement a string-matching algorithm.	C3 (Apply)	3,4
Course Description	**Topics**: Role of Algorithm in Computing, Analysis on nature of input and size of input , Designing Algorithms, Growth of Functions, Asymptotic Notations, Big-O, Big Ω, Big Θ, little-o, little-ω, loop invariants, Brute Force Approach, Divide-and-conquer approach; Sorting Algorithm analysis, Merge, Quick Sort, Greedy approach; Recursion and recurrence relations, Time Complexity of Recursive Algorithms, Algorithm Design Techniques, Dynamic Programming, Greedy Algorithms, String Matching, Search trees; Heaps; Hashing; Graph algorithms, sparse graphs, DFS, BFS, Minimum Spanning Trees, Shortest Path Algorithms, NP Completeness, Polynomial Time Algorithm, Polynomial Time verification.		
Text Book(s)	1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, 3rd Edition, The MIT Press, 2009, ISBN-10: 0262033844, ISBN-13: 978-0262033848		
Reference Material	1. Algorithm Design, (1st edition, 2013/2014), Jon Kleinberg, Eva Tardos 2. Algorithms, (4th edition, 2011), Robert Sedgewick, Kevin Wayne		