ICS211 Algorithms 3-1-0-4

Pre-Requisite: Discrete Mathematics and Data Structures.

Objective: To teach the basics of algorithm design and analysis.

Course Topics: (A) Thinking towards a solution (algorithm design): This objective is usually achieved by studying several exemplary techniques like: (a) Divide and Conquer. (b) Greedy Paradigm. (c) Dynamic Programming. (d) Linear Programming. (e) Backtracking. (f) Branch-and-Bound etc. **(B)** Analyzing the efficacy (correctness) and efficiency (feasibility) of purported solutions: This objective is usually achieved through several relevant examples like: (a) Solving recurrence relations. (b) Proving the correctness of a few number-theoretic algorithms. (c) Matroid theory (for greedy algorithms). (d) Proofs of correctness (usually via induction) for several optimization algorithms. (e) Basic results in probability theory (for randomized algorithms). **(C)** Insightfully appreciating the inherent complexity of a given problem: This objective is usually achieved by proving lower bound results and studying basic complexity classes like: (a) The Class P. (b) The Class NP. (c) The Class NPC. (d) The Class BPP. (e) The Class BQP. **(D)** Understanding the limits of computing: This objective is usually achieved by studying: (a) Godel's incompleteness theorem. (b) Church-Turing hypothesis. (c) Proving Undecidability via diagonalization/reduction. (d) Quantum Computing.
Preferred Text Books: (1). Cormen, Leiserson, Rivest, Stein. Introduction to Algorithms (3rd Ed), Prentice Hall of India. **(2).**Sipser Michael. Introduction to Theory of Computation, 2/e, Cengage Learnng.

Outcome: The students would learn how (a) to think about solving computation problems, (b) to prove the correctness and goodness of their solutions, (c) to continually attempt at improving on their approach and (d) to recognize if and when an optimal solution is designed.