DEPARTMENT OF INFORMATION TECHNOLOGY, NITK SURATHKAL

Course plan (Jul -Dec 2020)

Faculty in-charge	Sidney Rosario	**Year and Semester**	Jul-Dec 2020
Course Code:	IT 300	**Course Name:**	Design and Analysis of Algorithms
Core/Elective/MLC:	Program Core	**L-T-P:**	(3-0-2) 4
Pre-requisites:	IT202, IT251	**Contact Hours:**	**Jul-Dec** (Online Through IRIS) 3 per week - Lecture, 2 per week – Lab
Type of course: (Lecture/Tutorial/Seminar/Project)	Lecture and Lab	**Course Assessment Methods: (both continuous and semester-end assessment)**	**Theory: (60%)(online)** 20% Midsem Theory Exam 20% Online Quiz & Assignments 20% End Sem Theory **Lab: (40%)(Online)** 15% Continuous Evaluation 25% Course Minor Project

Course Description:

This course introduces students to the different paradigms and techniques used in solving algorithmic problems like Recursion, Divide and Conquer, Greedy approach, Dynamic Programming, Randomization, Heuristics and Approximation algorithms. Each of these paradigms will be taught using practical, real-world applications. Students will gain experience in discarding irrelevant information in a real-world problem, stripping it down to its mathematical essence in order to define the problem unambiguously, and then applying one or more of the above techniques to solve the problem. Together with designing and describing algorithms, students will also learn how to analyse & compare algorithms, in order to ensure their correctness and evaluate their performance.

Course Outcomes: On completing this course a student should be able

CO1[L1,L2,L3]: To understand the different tools & paradigms useful in designing algorithms.
CO2[L4] : To analyse the correctness and the efficiency of algorithms.
CO3[L5] : To compare and evaluate algorithms, and pick the one best suited for a given task.
CO4[L6] : To design & implement new algorithms, using the different techniques studied in the course.

Course Plan:

- Week 1 & 2 : Models of computation, algorithm analysis and asymptotic notation, time and space complexity, average and worst case analysis, lower bounds.
- Week 3,4: Recursion, Divide and Conquer: Applications to Integer multiplication, Selection Problem.
- Week 5,6: Greedy Algorithms: Applications in Job Scheduling, Data encoding, Cache replacement algorithms. Fractional Knapsack.
- Week 7,8: Dynamic Programming: Applications to Sequence Alignment, Matrix chain multiplication, Shortest Path algorithms.
- Week 9: Randomized Algorithms: Randomized Selection,Karger's Min-cut algorithm, Randomized algorithm for computing the closest pair of points.
- Week 10,11: Complexity classes, Introduction to the theory of NP-Completeness.
- Week 12: Introduction to Approximation Algorithms.
- Week 13: Local Search algorithms and Heuristics.

Text Books and References:

- Jon Kleinberg and Eva Tardos, "*Algorithm Design"*, 1st Edition, Pearson Education India, 2013.
- S Dasgupta, C Papadimitriou, U Vazirani, "*Algorithms*", McGraw-Hill Education, 2006.
- T H Cormen, C E Leiserson, R L Rivest, C Stein, "*Introduction to Algorithms"*, 3rd Edition, PHI, 2010.
- Steven S Skiena, "*The Algorithm Design Manual"*, 2nd Edition, Springer-Verlag, 2nd Edition, 2013.
- NPTEL Lectures on the "*Design and Analysis of Algorithms"*, by Dr. Abhiram Ranade and Dr. Sundar Vishwanathan, IIT-Bombay.
- Lectures on Algorithms by Dr. Tim Roughgarden on coursera.org.

Assessment CO matrix

Assessment Type	**Course Outcome (CO)**			
	CO1	**CO2**	**CO3**	**CO4**
Mid Sem Theory Exam	X	X	X	
Quizzes & Assignments	X	X	X	X
End Sem Theory Exam	X	X	X	
Lab Continuous Evaluations			X	X
Minor Project			X	X

Course Instructor

[Sidney Rosario]