COURSE CONTENT

ADVANCED ALGORITHMS								
II Semester: CSE								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
BCSE13	**Core**	3	0	0	3	40	60	100
Contact Classes: 45	**Total Tutorials: Nil**	**Total Practical Classes: Nil**				**Total Classes: 45**		
Prerequisites: Advanced Data Structures								

I. COURSE OVERVIEW:

This course typically aims to equip students with a deep understanding of fundamental algorithmic techniques, their analysis, and their applications in solving complex computational problems. This course includes graph theory, flow networks, and linear programming**.** Gain an understanding of a wide range of advanced algorithmic problems, and their application to real-world problems**.**

II. COURSE OBJECTIVES:

The students will try to learn:

I. The advanced methods of designing and analyzing algorithms.
II. The student should be able to choose appropriate algorithms and use it for a specific problem.
III. Students should be able to understand different classes of problems concerning their computation difficulties.

III. COURSE OUTCOMES:

After successful completion of the course, students will be able to:

CO1 Analyze algorithm efficiency, including time complexity, and space complexity for problem solving techniques
CO2 Determine the appropriate data structure for solving a particular set of problems
CO3 Develop algorithms using advanced techniques such as graph algorithms, and flow algorithms
CO4 Apply substitution, recurrence-tree methods to solve recurrences
CO5 Gain an understanding of a wide range of advanced algorithmics, and their application to real-world problems
CO6 Explore the complexity theory, and the concept of NP-completeness, and be able to identify and classify problems within different complexity classes

IV. COURSE CONTENT:

MODULE-I: Role of algorithms in computing (9)

Role of algorithms in computing, Analyzing algorithms, Designing Algorithms,
Growth of Functions, Divide and Conquer- The maximum-subarray problem, Strassen's algorithms for matrix multiplication, The substitution method for solving recurrences, The recurrence-tree method for solving recurrence, The master method for solving recursions, Probabilistic analysis, and random analysis

MODULE-II: Review of Data Structures (9)

Review of Data Structures- Elementary Data Structures, Hash Tables, Binary Search Trees, and Red-Black Trees.

MODULE-III: Elements of dynamic programming (9)

Elements of dynamic programming, - Matrix-chain multiplication, Longest common subsequence, Greedy Algorithms - Elements of the greedy strategy, Huffman codes, Amortized Analysis - Aggregate analysis, The accounting method, The potential method, Dynamic tables**.**

MODULE-IV: Flow – Networks (9)

Flow-Networks: Maxflow-min-cut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm.

MODULE-V: Shortest Path in Graphs (9)

Shortest Path in Graphs: Floyd-Warshall algorithm and introduction to dynamic programming paradigm. More examples of dynamic programming.
Linear Programming: Geometry of the feasibility region and Simplex algorithm
NP-completeness: Examples, proof of NP-hardness and NP-completeness.

V. TEXTBOOKS

1. Cormen, Leiserson, Rivest, Stein, "Introduction to Algorithms". The MIT Press, 4th edition, 2022.
2. Aho, Hopcroft, Ullman "The Design and Analysis of Computer Algorithms", Pearson Education, 7th edition, 2018.

VI. REFERENCE BOOKS:

1. Kleinberg and Tardos "Algorithm Design", Pearson Education, 2nd edition, 2016.

VII. WEB REFERENCES:

1. http://www.personal.kent.edu/~rmuhamma/Algorithms/algorithm.html
2. http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAlgorithms
3. http://www.facweb.iitkgp.ernet.in/~sourav/daa.html

VIII. E-TEXTBOOKS:

1. https://mitpress.mit.edu/9780262530910/
2. https://edutechlearners.com/download/Introduction_to_algorithms-3rd%20Edition.pdf
3. https://books.google.co.in/books/about/Introduction_To_Algorithms.html?id=NLngYyWFl_YC&redir_esc=y

IX. MATERIALS ONLINE

1. Course template
2. Tutorial question bank
3. Tech talk topics
4. Open,Ended experiments
5. Definitions and terminology

6. Assignments
7. Model question paper, I
8. Model question paper, II
9. Lecture notes
10. Power Point presentation
11. E-Learning Readiness Videos (ELRV)