MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act,1956

COURSE CONTENT

ADVANCED ALGORITHMS LAB								
II Semester: CSE								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
2425835	**Professional Core courses**	0	0	4	2	40	60	100
Contact Classes: 0	**Tutorial Classes: 0**	**Practical Classes: 60**				**Total Classes: 60**		
Prequisites: Strong foundation in Data Structures								

Course Overview:

The student can able to attain knowledge in advanced algorithms

Course Objectives:

1. To understand and implement advanced algorithm design techniques such as divide and conquer, greedy, and dynamic programming methods.
2. To develop practical skills in solving computational problems using advanced algorithms and optimization techniques.
3. To analyze the performance and complexity of algorithms through experimental implementation and testing.
4. To gain hands-on experience in implementing graph, string matching, and network flow algorithms.
5. To enhance problem-solving and programming abilities for designing efficient algorithms applicable to real-world scenarios.

Course Outcomes: After Completion of the Course, Students should be able to

1. Apply algorithmic techniques such as brute force, greedy methods, divide-and-conquer, and dynamic programming to implement solutions for classical problems like assignment, knapsack, an F71:F72d long integer multiplication.
2. Make use of numerical methods and matrix computations, including Gaussian elimination, LU decomposition, and Warshall's algorithm, to solve linear systems and determine transitive closures in graphs.
3. Analyze string-matching algorithms (Rabin–Karp, KMP, Horspool) and graph algorithms (max-flow) to determine correctness, efficiency, and computational complexity
4. Evaluate the performance trade-offs of different algorithmic approaches (brute force vs. greedy vs. divide-and-conquer) in terms of time, space, and suitability for specific problem scenarios.
5. Implement algorithms and measure their runtime behavior on sample inputs to compare with theoretical expectations.

LIST OF EXPERIMENTS

1. Implement assignment problem using Brute Force method
2. Perform multiplication of long integers using divide and conquer method.
3. Implement a solution for the knapsack problem using the Greedy method.
4. Implement Gaussian elimination method.
5. Implement LU decomposition
6. Implement Warshall algorithm
7. Implement the Rabin Karp algorithm.
8. Implement the KMP algorithm.
9. Implement Harspool algorithm
10. Implement max-flow problem

TEXT BOOKS:

1. Design and Analysis of Algorithms, S. Sridhar, OXFORD University Press

REFERENCE BOOKS:

1. Introduction to Algorithms, second edition, T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, PHI Pvt. Ltd./ Pearson Education.
2. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharam, Universities Press.
3. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson education

ELECTRONIC RESOURCES:

1. **https://nptel.ac.in/courses/106/101/106101060/**
2. **https://www.geeksforgeeks.org/fundamentals-of-algorithms/**
3. **https://www.coursera.org/specializations/algorithms**
4. **https://www.codechef.com/**

MATERIALS ONLINE:

1. Course template
2. Open-ended experiments
3. Definitions and terminology
4. Lab Manual