**COURSE PLAN**

# For

**DESIGN AND ANALYSIS OF ALGORITHMS (21CS138)**

Course Coordinator : Ms. Neelima Gurrapu

Course Instructors :

1. Dr B Vijaya Prakash 2. Dr M SureshKumar

3.Dr A Ramesh Babu 4. Dr A Vamshi

5.Johnson 6.CH Rajitha

7.D Ramakrishna 8 N.Shiva Prasad

9. Ch Aparna 10.M.Shylaja

11.R Sujitha 12.B Raju

13.Md AnkushavaliK 14.Swetha

15.M .Madhuri 16.G.Kranthi

17.K.Mamatha

Course Type : Professional Core

Semester and Year : V Semester and III Year

L-T-P : 3-0-4

Credits : 5

School : School of CS&AI

Department : Department of CS&AI

Course Level : UG

**School of**

**Computer Science & Artificial Intelligence**



SR University,

Warangal

## COURSE CONTEXT

|  |  |  |  |
| --- | --- | --- | --- |
| **SCHOOL** | CS & AI | **DEPARTMENT** | CS & AI |
| **DEGREE** | **B.Tech** | **DATE THIS COURSE WILL BE FFECTIVE FROM** | Jul–Dec,2024 |

**COURSE BRIEF**

|  |  |  |  |
| --- | --- | --- | --- |
| **COURSE TITLE** | Design And Analysis  Of Algorithms | **PRE-REQUISITES** | Data Structures,  Programming Language |
| **COURSE CODE** | 21CS138 | **TOTAL CREDITS** | 5 |
| **COURSE TYPE** | Professional Core | **L-T-P FORMAT** | 3-0-4 |

## COURSE SUMMARYs

To skill the students in understanding and applying the key techniques for designing of efficient algorithms. Students will be able to appreciate and analyze the concept of Computational complexity for time efficiency. It will also help them to understand the other major concepts necessary to work on computer algorithms.

## COURSE-SPECIFIC LEARNING OUTCOMES (CO)

At the end of the course the students will be able to

1. Analyse the asymptotic performance of algorithms.
2. Solve problems using key techniques of algorithm design.
3. Develop optimal solution by applying various methods and differentiate polynomial and non-polynomial problems.
4. Apply approximation algorithms for real time solutions.

|  |
| --- |
| **Course Articulation Matrix** |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | PO  CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | | CO1 | 3 | 3 | 2 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 3 | 2 | | CO2 | 3 | 3 | 3 | 2 | 3 | 1 | 1 | 2 | 2 | 2 | 1 | 2 | 3 | 3 | | CO3 | 3 | 3 | 3 | 3 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | | CO4 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | | Mapping Target Level | 3 | 3 | 2.75 | 2.5 | 2.25 | 1.25 | 1.25 | 2 | 1.75 | 2 | 1.5 | 2 | 3 | 2.75 | |

## DETAILED SYLLABUS

**UNIT I (7 Hours)**

Basics of Algorithms and Mathematics: - Introduction to an algorithm. The efficient algorithm, Average, Best and worst-case analysis of Time Complexity and Space Complexity. Asymptotic Notations. Sorting Algorithms and analysis: - Bubble sort, Selection sort, Insertion sort, sorting in linear time: Radix sort.

Divide and Conquer Algorithm: - Introduction, multiplying large Integers Problem.

**UNIT II (11 Hours)**

Problem Solving using divide and conquer algorithm: Binary Search, Max-Min problem, Sorting (Merge Sort, Quick Sort), Strassen’s Matrix Multiplication.

Greedy Algorithm: - General Characteristics, Problem solving using Greedy Algorithm- Activity selection problem. Minimum Spanning trees (Kruskal’s algorithm, Prim’s algorithm).

Graphs: - Shortest paths, The Knapsack Problem, Job Scheduling Problem.

**UNIT III (8 Hours)**

Dynamic Programming: - Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming– Matrix chain multiplication, All Pair Shortest path, 0/1 Knapsack, Optimal Binary Search Tree, Travelling Salesman Problem.

**UNIT IV (10 Hours)**

Backtracking: - Introduction, The Eight queen’s problem, sum of subsets.

Branch and Bound: 0/1 Knapsack Problem

String Matching: - Introduction, The naive string-matching algorithm, The Rabin Karp algorithm, The Knuth Morris-Pratt algorithm.

**For Lab Based Courses – Mention the experiments need to be covered in the lab**

The students will be implementing the fundamental design paradigms like dynamic paradigm, greedy algorithms, graphs traversing and several others. The preferred language of the course will be C. Furthermore, the students will be actively participating in open source projects available at GitHub.

**UNIT V(4 Hours)**

Approximation algorithms: - Travelling Salesman problem, Hamiltonian problem. Introduction to NP-Completeness: - The class P and NP, Polynomial reduction, NP Completeness Problem, NP-Hard Problems.

## STUDIO WORK / LABORATORY EXPERIMENTS:

The students will be implementing the fundamental design paradigms like dynamic paradigm, greedy algorithms, graphs traversing and several others. The preferred language of the course will be C. Furthermore, the students will be actively participating in open source projects available at GitHub.

## TEXTBOOKS/LEARNING RESOURCES:

1. Cormen, Leiserson, Rivest and Stein, Introduction to Algorithms (3rd ed.), The MIT Press, 2010. ISBN 978-0262033848.
2. Jon Kleinberg and Eva Tardos, Algorithm Design (1st ed.), Pearson, 2013. ISBN 978-9332518643.

## REFERENCE BOOKS/LEARNING RESOURCES:

1. V. Aho, J. E. Hopcroft and J. D. Ullman, Data Structures and Algorithms (1st ed.), Addison Wesley, Pearson,, 1983. ISBN 978-0201000238.
2. Narsimha Karumanchi, Algorithm Design Techniques: Recursion, Backtracking, Greedy, Divide and Conquer and Dynamic Program (1st ed.), Career Monk Publications, 2018. ISBN 978-8193245255.

## MOOC COURSES (Reference to the Course):

## 1. https://onlinecourses.nptel.ac.in/noc19\_cs47/preview

**Lecture Wise Plan**

|  |  |
| --- | --- |
| **No.** | **Content Planned** |
| 1 | Course structure/handout assessment mechanism (20),  Introduction to Algorithm (10),  Time Complexity and Space Complexity (25) |
| 2 | The efficient algorithm;  Asymptotic Notations: Big Oh, Theta, Omega (55) |
| 3 | Asymptotic Notations: Big Oh, Theta, Omega - Continued (55) |
| 4 | Sorting Algorithms and analysis: - Bubble sort, Selection sort (55) |
| 5 | Insertion sort (15), Radix Sort (40) |
| 6 | Radix Sort - Continued (45), Exercise Problem (10) |
| 7 | Divide and Conquer Introduction(20),Multiplying Large Integers Problem(35) |
| 8 | Problem Solving using Divide and Conquer - Binary Search (50) |
| 9 | Max-Min Problem (50) |
| 10 | Merge Sort (45), Exercise Problem (10) |
| 11 | Quick Sort (45), Exercise Problem (10) |
| 12 | Strassen's Matrix Multiplication (55) |
| 13 | Greedy Alogithm: General Characteristics(15),Activity Selection Problem(40) |
| 14 | Minimum Spanning Tree - Prim's Algorithm (55) |
| 15 | Minimmum Spanning Tree - Krushkal's Algorithm (55) |
| 16 | Single Sourse Shortest Path (55) |
| 17 | Knapsack Problem (55) |
| 18 | Job Scheduling Problem (55) |
| 19 | Dynamic Programming: Introduction (10),  Principle of Optimality (10),  0-1 Knapsack (35) |
| 20 | Matrix Chained Multiplication (55) |
| 21 | Matrix Chained Multiplication - continued (55) |
| 22 | All Pair Shortest Path (55) |
| 23 | Optimal Binary Search Tree (55) |
| 24 | Optimal Binary Search Tree - continued (55) |
| 25 | Travelling Salesman Problem (55) |
| 26 | Travelling Salesman Problem (55) |
| 27 | Backtracking: State-Space Search Tree (10),  N-Queen’s problem (45) |
| 28 | Sum of Subsets (55) |
| 29 | Sum of Subsets - continued (15),  Branch and Bound Introduction (40) |
| 30 | Buffer Lecture |
| 31 | 0/1 Knapsack (55) |
| 32 | 0/1 Knapsack - Continued (55) |
| 33 | String Matching Algorithms: Introduction (55) |
| 34 | Naive string matching Algorithm (55) |
| 35 | Rabin Karp Algorithm (55) |
| 36 | Knuth Morris-Pratt algorithm (55) |
| 37 | Knuth Morris-Pratt algorithm - Continued (55) |
| 38 | Approximation algorithms: Travelling Salesman problem (55) |
| 39 | Hamiltonian Problem (55) |
| 40 | Introduction to NP-Completeness: - P and NP, NP Complete and NP-Hard (55) |
| 41 | Buffer Lecture |
| 42 | Buffer Lecture |

**Lab Wise Plan**

|  |  |
| --- | --- |
| **No.** | **Content Planned** |
| **1** | Implement Bubble Sort, Selection Sort and Insertion Sort |
| 2 | Radix Sort |
| 3 | Binary Search and Merge Sort |
| 4 | Quick Sort |
| 5 | Multiplying large Integers Problem |
| 6 | Max-Min problem using Divide and Conquer,  Fractional Knapsack Using GP |
| 7 | Activity Selection Using GP  Minimum Spanning Tree: Prim’s Algorithm |
| 8 | Minimum Spanning Tree: Kruskal’s Algorithm  Dijkstra's Algorithm |
| 9 | 0-1 Knapsack Using DP  All Points Shortest Path Using DP |
| 10 | Matrix Chain Multiplication using DP |
| 11 | Travelling Salesman Problem |
| 12 | N-Queen  Sum of Subset |
| 13 | Knuth Morris –Pratt algorithm |
| 14 | Hamiltonian problem |

**Evaluation Components**

|  |  |
| --- | --- |
| **Components of Course Evaluation** | **Percentage** |
| Mid Term Examination | 20 |
| End Term Examination | 40 |
| Continuous Lab Evaluation | 20 |
| Continuous Tutorial Evaluation | - |
| Lab Exam | 10 |
| Quiz | - |
| Certification | 10 |
| Project | - |
| Any Other Assessment | - |

* Proposed Industry Talks: 02
* Start-ups related to the Course: --
* Certification Mapping: --
* Software/Tools Used: Google colab
* Hardware/Devices Used:
* Proposed Case Studies:
* Advanced Research Topics:

**Attendance Policy**

# At least 75% attendance in the course is mandatory

1. A maximum of 5% shall be allowed under medical grounds and 5% on representing the University on official events outside like sports, hackathons, NCC, NSS etc.

# Students with less than 65% of attendance shall be prevented from writing the final assessment.

**Academic Dishonesty & Plagiarism**

Plagiarism is "to offer work or ideas from another source as one's own, with or without authorization of the source author(s), directly by verbatim copying or by usage of any AI software" (i.e., with or without permission from the original author). In certain cases, authorization might be provided for the usage of other sources through written permission may not be considered as plagiarism. It is a serious academic offence which should be avoided, the following method will be adopted to evaluate plagiarism in submitted documents including assignments, material, class test content and other similar academic documents.

Level 1: Similarities up to 20% - Student will be asked to revise the document and resubmit for evaluation, once chance will be provided to revise.

Level 2: Similarities above 20% to 40%- Student will be warned and one chance will be provided to revise the document and resubmit.

Level 3: Similarities above 40% and above: If the plagiarism level is more than 40% student will get a Fail grade.

**Instructor Responsible for Lecture PPTs Preparation**

1. Dr.M.SureshKumar

**Instructor Responsible for Preparation of Lab Assignments**

1**.**  Dr A Vamshi

2. Johnson