Course: BTech **Semester:** 5

Prerequisite: Strong programming skills and a solid understanding of algorithms and their analysis are prerequisites for learning and applying Design and Analysis of Algorithms | 203105101 - Fundamentals of Programming

Course Objective: Design and Analysis of Algorithms (DAA) is crucial for efficient problem-solving and algorithm development. It provides tools to measure algorithm performance and make informed decisions on choosing the best algorithms for specific tasks. DAA helps optimize time and space complexities, leading to improved computational efficiency.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
0	0	4	0	2	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:

1. Develop the ability to design and implement efficient algorithms for fundamental problems.
2. Cultivate critical thinking skills to analyze problem requirements and constraints, allowing for the selection and modification of appropriate algorithms to solve specific computational problems.
3. Master the use of essential data structures such as arrays, matrices, graphs, and trees to efficiently store, manage, and manipulate data within algorithm implementations.
4. Learn techniques for optimizing algorithms to improve their efficiency and scalability, focusing on aspects such as time complexity, and space complexity,

List of Practical

1.	**write a program to determine whether the given number is Prime or not.**
2.	**Given a sorted array and a target value,return the index if the target is found. If not,return the index where it would be if it were inserted in order.**
3.	**There are N children standing in a line with some rating value. You want to distribute a minimum number of candies to these children such that: Each child must have at least one candy. The children with higher ratings will have more candies than their neighbours. You need to write a program to calculate the minimum candies you must give.**
4.	**There is a new barn with N stalls and C cows. The stalls are located on a straight line at positions x1,xN (0 <= xi <= 1,000,000,000). We want to assign the cows to the stalls, such that the minimum distance between any two of them is as large as possible. What is the largest minimum distance?**
5.	**Given an undirected graph with V vertices and E edges, check whether it contains any cycle or not**
6.	**There are n servers numbered from 0 to n – 1 connected by undirected server-to-server connections forming a network where connections[i] = [ai, bi] represents a connection between servers ai and bi. Any server can reach other servers directly or indirectly through the network.A critical connection is a connection that, if removed, will make some servers unable to reach some other servers. Return all critical connections in the network in any order.**
7.	**Given a grid of size NxM (N is the number of rows and M is the number of columns in the grid) consisting of '0's (Water) and '1's(Land). Find the number of islands.**
8.	Given a grid of dimension N x M where each cell in the grid can have values 0, 1, or 2 which has e following meaning: 0: Empty cell 1: Cells have fres 2. Cells have oranrotten oranges We have to determine what is the minimum time required to rot all oranges. A rotten orange at index [i,j] can rot other fresh oranges at indexes [i-1,j], [i+1,j], [i,j-1], [i,j+1] (up, down, left and right) in unit time'
9.	**Given two strings str1 and str2 and below operations that can be performed on str1. Find minimum number of edits (operations) required to convert 'str1' into 'str2'. Insert Remove Replace, All of the above operations are of equal cost.**
10.	**Minimum Path Sum" says that given a n x m grid consisting of non-negative integers and we need to find a path from top-left to bottom right, which minimizes the sum of all numbers along the path.**
11.	**Given string num representing a non-negative integer num, and an integer k, return the smallest possible integer after removing k digits from num.**
12.	**There is a robot on an m x n grid. The robot is initially located at the top-left corner (i.e., grid[0][0]).The robot tries to move to the bottom-right corner (i.e., grid[m - 1][n - 1]). The robot can only move either down or right at any point in time.Given the two integers m and n, return the number of possible unique paths that the robot can take to reach the bottom-right corner.**

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc

Course: BTech **Semester:** 5

Prerequisite: Data structures, Fundamental of programming

Course Objective: Analyze the asymptotic performance of algorithms. Write rigorous correctness proofs for algorithms. Demonstrate a familiarity with major algorithms and data structures. Apply important algorithmic design paradigms and methods of analysis. Synthesize efficient algorithms in common engineering design situations.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction and Analysis of Algorithms:** Algorithm: Definition, Properties, Types of Algorithms, Writing an AlgoritAlgorithm Analysis: Parameters, Design Techniques of Algorithms Asymptotic Analysis: Big Oh, Big Omega & Big Theta Notations, Lower Bound, Upper Bound and Tight Bound, Best Case, Worst Case, Average Case Analyzing control statement, Loop invariant and the correctness of the algorithm, Recurrences- substitution method, recursion tree method, master method. Sorting Techniques with analysis: Bubble Sort, Selection Sort, Insertion sort.	20	10
2	**Divide & Conquer Algorithms:** Structure of divide-and-conquer algorithms, examples: Binary search, quick sort, Merge sort**,** Strassen Multiplication; Max-Min problem	20	6
3	**Greedy Algorithms:** Introduction, Elements of Greedy Strategy - Minimum Spanning Tree: Kruskal's & Prim's Algorithm, Dijkstra's Algorithm, Knapsack Problem, Activity Selection Problem, Huffman Codes	20	8
4	**Dynamic Programming:** Principal of Optimality, 0/1 Knapsack Problem, Making Change problem, Chain matrix multiplication, Longest Common Subsequence, All pair shortest paths: Warshall's and Floyd's algorithms	20	8
5	**Exploring Graphs:** An introduction using graphs and games, Undirected Graph, Directed Graph, Traversing Graphs, Depth First Search, Breath First Search, Topological sort	5	3
6	**Backtracking and Branch & Bound:** Introduction to Backtracking, Introduction to Branch & Bound, 0/1 Knapsack Problem, N-Queens Problem, Travelling Salesman Problem	5	4
7	**String Matching & NP Completeness:** **String Matching:** **-** Introduction to String Matching, Naive String Matching, Rabin-Karp Algorithm, Kruth-Morris-Pratt Algorithm, String Matching using Finite Automata **NP Completeness:** - Introduction to NP Completeness, P class Problems, NP Class Problems, Hamiltonian Cycle	10	6

Reference Books

1.	**Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill. (TextBook)**
2.	**Fundamentals of Algorithms – E. Horowitz et al. (TextBook)**
3.	**Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson**
4.	**Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.**
5.	**Algorithms—A Creative Approach,3RD Edition, UdiManber, Addison-Wesley, Reading, MA**

Course Outcome

After Learning the Course the students shall be able to:

Course Outcome: After learning the course the students will be able to:
1. Develop the ability to analyze the running time of any given algorithm using asymptotic analysis and prove the correctness of basic algorithms.
2. Design efficient algorithms for computational problems, using various algorithm design techniques taught in the course.
3. Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate.
4. Analyze String matching algorithms.
5. Explain the complexity classes P, NP, and NP-Complete, and demonstrate the NP-Completeness of a specific problems.

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc