**22AIE212 DESIGN AND ANALYSIS OF ALGORITHMS L-T-P-C: 2- 0- 3- 3**

# Course Objectives

* This course helps students to impart various design techniques for formulation of algorithm.
* This course helps students to understand basic categories of algorithms.
* This course helps students to understand and apply analysis of space and time complexity of algorithms and understand concept of growth rate.
* This course helps students to deliver standard notations and representations of algorithmic complexity and known complexities.
* This course helps students to comprehend basic complexity classes.
* This course helps students to acquaint with will know tractable and intractable problems and map solutions to it.

# Course Outcomes

After completing this course, the students will be able to

**CO1:** Develop skills for analyzing algorithmic strategies.

**CO2:** Apply appropriate algorithmic technique for a given problem.

**CO3:** Implement standard algorithms on arrays, strings, trees and graph.

**CO4:** Analyse the nature of known classes of tractable or intractable problem.

# CO-PO Mapping

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| CO |
| CO1 | 3 | 3 | 3 | 3 | 3 | 3 | 1 | - | 3 | 3 | 2 | 3 | 3 | 1 | - |
| CO2 | 3 | 3 | 3 | 2 | 3 | 2 | - | - | 3 | 3 | 2 | 3 | 3 | 2 | - |
| CO3 | 3 | 3 | 3 | 3 | 2 | 1 | - | - | 3 | 3 | 3 | 3 | 3 | 3 | - |
| CO4 | 3 | 3 | 3 | 3 | 2 | 1 | - | - | 3 | 3 | 3 | 3 | 2 | 3 | - |

**Syllabus Unit 1**

Notion of an Algorithm – Fundamentals of Algorithmic Problem Solving – Important Problem Types – Fundamentals of the Analysis of Algorithmic Efficiency –Asymptotic Notations and growth rate- Empirical analysis – Recursive and non-Recursive Templates. Brute Force: Exhaustive Search and String Matching, Divide and Conquer Methodology: Binary Search – Merge sort – Quick sort – Heap Sort – Multiplication of Large Integers.

# Unit 2

Dynamic programming: Principle of optimality – Coin changing problem, Computing a Binomial Coefficient – Floyd‘s algorithm – Multi stage graph – Optimal Binary Search Trees – Knapsack Problem and Memory functions. Greedy Technique: Container loading problem – Huffman Trees. Iterative methods: The Simplex Method – The Maximum-Flow Problem – Maximum Matching in Bipartite Graphs, Stable marriage Problem, Measuring Limitations: Lower – Bound Arguments – P, NP, NP- Complete and NP Hard Problems.

# Unit 3

Backtracking – n-Queen problem – Hamiltonian Circuit Problem – Subset Sum Problem, Branch and Bound – LIFO Search and FIFO search – Assignment problem – Knapsack Problem – Travelling Salesman Problem, Approximation Algorithms for NP-Hard Problems – Travelling Salesman problem – Knapsack problem revisited.

# Textbooks/References

*Jeffrey McConnell, Analysis of algorithms. Jones & Bartlett Publishers, 2nd Revised edition, 2007.*

*Anany Levitin, Introduction to the Design and Analysis of Algorithms, Third Edition, Pearson Education, 2012. Harsh Bhasin, Algorithms Design and Analysis, Oxford university press, 2016*

Amrita Vishwa Vidyapeetham. BTC-AIE B.Tech Curriculum June 2022