Amrita School of Computing, Amritapuri Campus
Amrita Vishwa Vidyapeetham
Department of Computer Science and Engineering
Course Plan April-August 2024

22AIE 212 DESIGN AND ANALYSIS OF ALGORITHMS(L-T-P-C:2-0-3-3)
S4 B.Tech CSE(AIE)

Faculty Information

Name : **Dr.Remya. S**
Assistant Professor, ASC, Amritapuri
Email : **remyas@am.amrita.edu**

Contact No: 9447746568

Course Description:

This course offers a comprehensive exploration of algorithmic problem-solving techniques and analysis, divided into three units. Starting with fundamental notions of algorithms and efficiency analysis, students delve into key problem types such as brute force, divide and conquer, dynamic programming, greedy techniques, and backtracking. Through practical applications and theoretical concepts, including asymptotic notations, students gain proficiency in algorithm design and analysis, enabling them to tackle a wide range of computational challenges efficiently.

Prerequisites:
22AIE112 Data Structures & Algorithms-1
22AIE203 Data Structures and Algorithms-2

Course Learning Outcomes (CO):

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 Affinity Map

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
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	-

Course 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

[R1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. Introduction to Algorithms, Third Edition (3rd ed.). The MIT Press.

[R2] Anany Levitin, Introduction to the Design and Analysis of Algorithms, Third Edition, Pearson Education, 2012.

[R3] Jeffrey McConnell, Analysis of algorithms. Jones & Bartlett Publishers, 2nd Revised edition, 2007.

[R4] Harsh Bhasin, Algorithms Design and Analysis, Oxford university press, 2016

Lecture Plan

Week #	Lecture No(s)	Topics	CO	References
Week#1	1	Course Introduction, Course plan, Algorithm definition, properties of algorithm, How to write algorithm	1	R1 - 1.1, 2.1
	2	Algorithm analysis framework-input size, running time-order of growth rate, RAM Machine/Model	1	R1 - 2.2
Week#2	3	Analysis framework - Input size, basic operation, rate of growth of function , L′Hôpital′s Rule:, Examples	1	R2 - 2.1
	4	Asymptotic Analysis, Guidelines for asymptotic analysis, Asymptotic Notations - 1	1	R1 - 3.1 R2 - 2.2
Week #3	5	Asymptotic Notations - 2, basic efficiency classes, properties of asymptotic notation- comparison of functions	1	R1 - 3.1 R2 - 2.2
	6	Analysis of Iterative Algorithms-Binary Search	1,2	R2 - 2.3
Week #4	7	Analysis of recursive algorithms - Forming recurrence relation, solving recurrence relation - iterative method	2	R1 - 4.0, R2 - 2.4
	8	Recurrence tree	1,2	R1 - 4.4
Week #5	9	Master method	1	R1 - 4.5
	10	Substitution method	1	R1 - 4.3
Week #6	11	Divide and Conquer Methodology– Merge sort	2	R1 - 4 R2 - 4
	12	Quick sort	2	R2 - 4.2, R1 - 7.1
Week #7	13	Heap Sort	2	R2 - 4.2, R1 - 7.1
	14	Revision		
Week #8	Mid Term Examination			
Week #9	15	Multiplication of Large Integers , Dynamic programming: Principle of optimality – Coin changing problem,	3,4	R2 - 9, R1 -16.1
	16	Computing a Binomial Coefficient	3,4	R1-15.4

Week #10	17	Floyd‘s algorithm – Multi stage graph –Optimal Binary Search Trees	2	R1 - 25.2
	18	0-1 Knapsack	3,4	R2 - 8.4
Week #11	19	Matix chain multiplication	3,4	R1 - 15.2
	20	Greedy Technique: Huffman Trees.	3,4	R2 - 9, R1 -16.1
Week #12	21	Fractional Knapsack	3,4	R1 - 16.2,16.3
	22	Applications of Graphs- topological sort, cycles in directed graph	3,4	R1 - 22.4
Week #13	23	Biconnected components	3,4	R1 - 22.5
	24	Strongly connected Components	3,4	R1 - 22.5
Week #14	25	Network Flow - Ford-Fulkerson algorithm	3	R1 - 26.2
	26	Maximum Bipartite matching	3	R1 -26.3
Week #15	27	Introduction to NPC, definitions, examples	4	R1 -34
	28	Polynomial time Reduction - Hamiltonian path to cycle,3CNF to Clique	4	R1 - 34.1, 34.2, 34.3, 34.5.1
Week #16	29	Clique to vertex cover	4	R1 -34.5.2
	30	Backtracking – n-Queen problem	3,4	R2 - 12.1
Week #17	31	Subset Sum Problem	2,3	R1 - 4.1
	32	Branch and bound for 0-1 knapsack Problem	3,4	R2 - 12.2
Week #18	33	Travelling Salesman Problem	3,4	R1-35.2
	34	Revision		
Week #19	35	Revision		
	36	Revision		

Evaluation Policy: **70 (Internal Assessment) + 30 (End Semester)**

<table>
<tr><th>Evaluation Policy</th><th>Components</th><th>Remarks-Submissions</th><th>Split up</th><th>Weightage</th></tr>
<tr><td rowspan="4">Continuous Evaluation (Lab +Theory)</td><td>Quiz-total 2</td><td>Quiz #1-week 3
Quiz #2-Week 6</td><td>Quizzes with equal weightage (2*10M=20 Marks)</td><td rowspan="4">50 %</td></tr>
<tr><td>Assignment 1(followed by quiz)</td><td>Numerical Problems</td><td>1*5M=5 Marks</td></tr>
<tr><td rowspan="3">Assignment 2</td><td>Lab Sheet Evaluation + 1contest- Hacker Rank/ Leet code</td><td>5 Marks</td></tr>
<tr><td>Lab Viva (#1)</td><td>10 Marks</td></tr>
<tr><td>Lab Exam</td><td>10Marks</td></tr>
<tr><td>Mid Term</td><td colspan="3"></td><td>20 %</td></tr>
<tr><td>End Semester</td><td colspan="3"></td><td>30 %</td></tr>
</table>

Dr. Remya S
Assistant Professor
Department of CSE
Amrita Vishwa Vidyapeetham