School of Computer Engineering

Kalinga Institute of Industrial Technology (KIIT)
Deemed to be University
Bhubaneswar-751024

Lesson Plan and Activity Calendar

Design and Analysis of Algorithms – CS30001 (L-T-P-Cr: 3-0-0-3)

Semester: 5th
Discipline: B.Tech. (CSE), **Section:** CSE 47
Session: Autumn 2024

Prerequisites: Data Structures (CS21001)

Instructor:

Name	: Dr. Dayal Kumar Behera
Chamber	: Faculty Block 402, Cubicle –Z, Block-C, Campus-14
Mobile No	: 9853334495
Email	: dayal.beherafcs@kiit.ac.in

Class Hours:

Day	Time	Class Room
WED	08.00 – 09.00 AM	A-LH-104
THUR	11.00 – 12.00 PM	WL-103
FRI	09.00 – 10.00 AM	A-LH-110

:---

Lesson Plan:

Unit	Unit Name	Topics to be covered	No. of lectures	Lecture serial nos.
1	Introduction	● Concepts in algorithm analysis & design - motivation ● Complexity of an algorithm (Space and time Complexity), Analysis of time complexity of Insertion Sort by step count method ● Growth of functions, Asymptotic Notations (Big Oh, Omega, Theta)) ● Solving recurrences: Iterative method Substitution method, Recurrence Tree method ● Solving recurrences: Master theorem, Change of variable	5	1-5
		Tutorials / Activity		

2a	**Divide and Conquer Approach**	● Structure of Divide-and-Conquer algorithm design technique ● Analysis of divide-and-conquer run time recurrence relations of ■ Binary Search ■ Merge Sort ■ Quick Sort	**3**	**6-8**
		Tutorials / Activity		
2b	**Greedy Approach**	● Overview of Greedy design paradigm and Solving as well as analyzing the following problems using Greedy method: ■ Fractional knapsack problem ■ Job sequencing with deadlines ■ Huffman method of Optimal Coding ■ Finding Minimum spanning trees for a Graph: Kruskal's Method ■ Finding Minimum spanning trees for a Graph: Prim's Method ■ Finding Single Pair Shortest Path in a graph: Dijkstra's Method	**6**	**9-14**
		Tutorials / Activity		
3	**Dynamic Programming Approach**	● Overview of Dynamic Programming paradigm, Difference between Dynamic Programming and Divide & Conquer/Greedy Methods ● Solving as well as analyzing the following problems using Dynamic Programming method: ■ 0/1 Knapsack problem ■ Matrix Chain Multiplication ■ Longest Common Subsequence ● Multistage Graph problem - Forward and Backward Approach for solving ● Finding All Pair Shortest Paths in a Graph - Floyd Warshall Algorithm ● Notion of Optimal Binary Search Tree (OBST), Finding OBST using Dynamic Programming method ● Notion of Travelling Salesman Problem (TSP), Solving TSP using Dynamic Programming method	**10**	**15-24**
		Tutorials / Activity		
4	**Amortized Analysis and Randomized Algorithms**	● Notion and methods of Amortized Analysis of algorithms: Aggregate Analysis, Accounting Method, Potential Method ● Randomized Algorithms: Las Vegas and Monte Carlo algorithms ● Randomized Quick Sort and its analysis ● Minimum Cut in a graph - Karger's min cut algorithm	**6**	**25-30**

		Tutorials / Activity		
6.	**Complexity Classes and Approximation Algorithms**	● Concepts of Complexity Classes: P, NP, NP-Hard and NP-Complete ● Reducibility of problems ● Complexity Classes for the following selected problems: ■ Vertex Cover Problem ■ 3-CNF Satisfiability Problem ■ Maximal Clique Problem ■ Hamiltonian cycle Problem ■ Travelling Salesman Problem ● Notion of Approximation algorithms ● Approximation algorithms for following selective problems: ■ Vertex Cover Problem ■ Travelling Salesman Problem	**10**	**31-40**
		Tutorials / Activity		

Day-wise Lesson Handouts:

Week	Lecture No.	Topics
Week - 1	1	Concepts in algorithm, difference between Algorithm and Program, characteristics of algorithms. Algorithm design and their Performance Analysis: Time and Space Complexity, motivation.
	2	Pseudo code Conventions, Analysis of Insertion Sort by step count method (Incremental Approach), Best-case, Worst-case and Average-case Analysis.
	3	Growth of functions, Asymptotic Notations ($\theta,\ O,\ \Omega$)
Week - 2	4	Solving recurrences using Iterative method, Substitution method and Recursion Tree method
	5	Solving recurrences using Master's Theorem, Change of Variables
	6	Divide-and-Conquer Approach - Binary Search and its complexity analysis
Week – 3	7	Divide-and-Conquer Approach - Merge Sort and its complexity analysis
	8	Divide-and-Conquer Approach - Quick Sort and its complexity analysis (Worst-case, Best-case and role of Balanced Partitioning)
	9	Overview of Greedy paradigm, Elements of greedy strategy: Recursive, Iterative greedy algorithm. Knapsack Problem, Difference between Fractional Knapsack and 0/1 Knapsack, Greedy strategy for solving Fractional Knapsack Problem.
Week - 4	10	A greedy approach for Job sequencing with deadlines
	11	Problem of Optimal coding, Huffman Tree and Huffman method for Optimal coding
	12	Minimum-Cost Spanning Tree (MST) for a weighted graph, Kruskal's method for finding MST, its analysis
Week - 5	13	Prim's Method for finding MST, its analysis
	14	Single-Source Shortest Path problem for weighted graphs, Dijkstra's Method and its analysis
	15	Overview of Dynamic Programming paradigm, Divide and Conquer vs Dynamic Programming, Greedy vs Dynamic Programming, Elements of dynamic programming, Tabulation vs Memoization

Week - 6	16	Dynamic Programming approach for solving 0/1 Knapsack problem
	17	Longest Common Subsequence (LCS) problem, Dynamic Programming approach for solving LCS problem
	18	Matrix Chain Multiplication (MCM) problem, Dynamic Programming approach for solving MCM problem
Week - 7	19	Multistage Graph problem - Forward Approach
	20	Multistage Graph problem - Backward Approach
	21	All Pair Shortest Path problem for weighted graph - Floyd Warshall’s algorithm
Week - 8	22	Floyd Warshall’s algorithm (Contd)
	23	Notion of Optimal Binary Search Tree (OBST), Finding OBST using Dynamic Programming method
	24	Notion of Travelling Salesman Problem (TSP), Solving TSP using Dynamic Programming method
Week - 9	25	Amortized Analysis of algorithm - Aggregate Analysis
	26	Amortized Analysis of algorithm - Accounting Method
	27	Amortized Analysis of algorithm - Potential Method
Week - 10	28	Randomized Algorithms: Las Vegas and Monte Carlo
	29	Randomized Quick Sort and its analysis
	30	Minimum Cut in a graph, Karger’s min cut algorithm
Week - 11	31	Tractable vs Intractable problems, Decision vs Optimization Problems, Deterministic vs Nondeterministic Algorithms, Reduction, Polynomial Reduction and Equivalence of Problems.
	32	Basic Concepts of Complexity Classes (P, NP, NP hard, NP Complete) and their hierarchy
	33	SAT, CNF-SAT, 3CNF-SAT: Basic Concepts and Complexity Classes
Week - 12	34	Maximal Clique Problem
	35	Vertex Cover Problem
	36	Hamiltonian cycle Problem,
Week - 13	37	Travelling Salesman Problem
	38	Approximation Algorithm: Basic Concepts, Relation with NP hardness
	39	Approximation Algorithm for Vertex Cover Problem
	40	Approximation Algorithm for Travelling Salesman Problem

Activity Calendar:

Activity No.	Type of Activity	Probable Date	Marks (Weightage)	CO
1	**ACTIVITY-1** (Class Test/ Subjective Test)	01.08.24 – 06.08.24	5	
2	**ACTIVITY-2** (Class Test/Quiz Test)	19.08.24 – 23.08.24	5	
3	**ACTIVITY-3** (Class Test/ Surprise Test)	09.09.24 – 13.09.24	5	
Mid Semester Examination [17.09.2024 – 21.09.2024]				
4	**ACTIVITY-4** (Home Assignment)	01.10.24 – 07.10.24	5	
5	**ACTIVITY-5** (Class Test/Home Assignment)	21.10.24 – 25.10.24	5	
6	**ACTIVITY-6** (Class Test/Quiz Test)	04.11.24 – 08.11.24	5	
End Semester Examination [16.11.2024 – 26.11.2024]				

Course Outcome: Upon completion of this course, the students will be able to:

CO1:	Analyze the time and space complexity for any algorithm
CO2:	Compare and contrast different algorithm design techniques
CO3:	Apply the algorithm design techniques in solving real world problems
CO4:	Perform amortize analysis for any algorithm
CO5:	Modify existing algorithms to apply in common engineering design situations
CO6:	Use NP class of problems to propose approximation algorithms

Text books:

- Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, Universities Press.
- Thomas H. Corman, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, “Introduction to Algorithms”, PHI.

Reference books:

- Jon Kleinberg, Eva Tardos, “Algorithm Design”, Pearson.
- Michael T. Goodrich, Roberto Tamassia, “Algorithm Design: Foundations, Analysis, and Internet Examples”, Wiley India.

Grading Policy:

Pedagogy: Lecture, Assignments, Quiz, Debate, Short Projects, etc.

Evaluation Methodology: Internal: 50 (20- Midterm Exam & 30 Activity), End Term: 50

Distribution of Marks:

SL No.	Evaluation Component	Evaluation Marks	Course Lecture No.		Mode
			From	To	
1	Mid-Semester Examination	20	1	20	Closed Book
2	Activity based Teaching and Learning	30	NA	NA	Open Book, Closed Book and Presentation, Short quiz
3	End-Semester Examination	50	1	40	Closed Book

Note

- ***Tentative Mid-Semester Syllabus would be up to Longest Common Subsequence problem under the unit of Dynamic Programming approach as per the Lesson Plan***
- ***Modifications to the above-mentioned structure (Lesson Plan / Examination Process / Any other modifications) may take place as per the Teacher's discretion adhering to the University Guidelines.***

======================XXXXXXXX===================