A. Course Handout (Version 1.0)

Institute/School Name	Chitkara University Institute of Engineering and Technology		
Department Name	Department of Computer Science & Engineering		
Programme Name	Bachelor of Engineering (B.E.), Computer Science & Engineering		
Course Name	Advanced Data Structures	Session	2024-2025
Course Code	22CS036	Semester/Batch	6th/2022
L-T-P (Per Week)	2-0-4	Course Credits	04
Pre-requisite	Data Structures and Algorithms	NHEQF Level	06
Course Coordinator	Dr. Suhasini	SDG Number	4,8,9

CLO01	Understand the detailed view of data structures and algorithms with underlying mathematics behind it.
CLO02	Revisit Object Oriented fundamentals along with the concepts of other linear data structures like Linked lists and Stacks and nonlinear data structures like Graphs, Tries, Binary Trees and its variations; with main emphasis on Interview based questions.
CLO03	Explore various algorithm strategies such as DP, Greedy Method, Backtracking and Bit-masking.
CLO04	Analyze and evaluate different data structures and will be able to prepare well for Interview panels through numerical understanding of the concepts.
CLO05	Implement the concepts of data structures and algorithms on several forums like code-chef, coding ninjas, GFG and Hacker Rank.

1. Objectives of the Course

The course provides a wide scope of learning & understanding of the subject and the main objectives of the course are:

- To understand the detailed view of Arrays, Strings, Recursion, Backtracking.
- To learn object-oriented basics with strong up-skilling on various linear and nonlinear data structures.
- To explore and implement various algorithm design strategies using examples.
- To analyze and evaluate different data structures.
- To implement the concepts of data structures and algorithms by solving complex engineering problems and preparing well for interviews, competitions and hackathons.

2. Course Learning Outcomes: After completion of the course, student should be able to:

	Course Learning Outcome	*POs	**CL	***KC	Sessions
CLO01	Understand the detailed view of data structures and algorithms with underlying mathematics behind it.	PO1,PO2,PO3, PO4, PO9, PO11, PO12	K2	Factual, Conceptual	24
CLO02	Revisit Object Oriented fundamentals along with the concepts of other linear data structures like Linked lists and Stacks and nonlinear data structures like Graphs, Tries, Binary Trees and its variations; with main emphasis on Interview based questions.	PO1,PO2,PO3, PO4, PO9, PO11, PO12	K3	Conceptual, Procedural	30
CLO03	Explore various algorithm strategies such as DP, Greedy Method, Backtracking and Bit-masking.	PO1,PO2,PO3, PO4, PO9, PO11, PO12	K3	Conceptual, Procedural	30

CLO04	Analyze and evaluate different data structures and will be able to prepare well for Interview panels through numerical understanding of the concepts.	PO1,PO2,PO3, PO4, PO9, PO11, PO12	K4	Conceptual, Procedural	20
CLO05	Implement the concepts of data structures and algorithms on several forums like code-chef, coding ninjas, GFG and Hacker Rank.	PO1,PO2,PO3, PO4, PO9, PO11, PO12	K3	Conceptual, Procedural	8
	Total Contact Hours				112

Revised Bloom's Taxonomy Terminology

* PO's available at (shorturl.at/cryzF)

**Cognitive Level =CL

***Knowledge Categories = KC

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	H	H	M	H	M		M				L	
CO2	H	M			H							
CO3	H	H	M		H	L						
CO4	M											
CO5	M	M	M	L	M		L		L			

H=High, M=Medium, L=Low

3. ERISE Grid Mapping

Feature Enablement	Level (1-5, 5 being highest)
Entrepreneurship	2
Research	4
Innovation	3
Skills	5
Employability	5

4. Recommended Books:

B01. Computer Algorithms by E. Horowitz, S. Sahni and S. Rajsekran, Computer Science Press, New York, ISBN – 0-7167-8316-9.

B02. Introduction to Algorithms, Second Edition, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, MIT Press, Cambridge, Massachusetts London, England McGraw-Hill Book Company, ISBN 0-262-03293-7.

B03. Data Structures with C (Schaum's Outline Series) (English, Paperback, Lipschutz Seymour), McGrawHill Education India, ISBN: 9780070701984, 9780070701984

B04. Design & Analysis of Computer Algorithms (English, Paperback, Aho Alfred V.), Pearson Education India, ISBN: 9788131702055, 9788131702055

B05. Data Structures and Algorithms Made Easy (English, Paperback, Karumanchi Narasimha), Karumanchi Narasimha, Careermonk Publications, ISBN: 9788193245279, 9788193245279.

5. Other readings and relevant websites:

S.N.	Link of Journals, Magazines, Websites, and Research Papers
1.	https://onlinecourses.nptel.ac.in/noc22_cs26/preview
2.	https://www.youtube.com/watch?v=zWg7U0OEAoE
3.	https://iq.opengenus.org/list-of-advanced-data-structures/
4.	https://in.coursera.org/learn/advanced-data-structures

5.	https://www.geeksforgeeks.org/advanced-data-structures/
6.	https://www.youtube.com/@JennyslecturesCSIT/videos

6. Recommended Tools and Platforms

Testpad, Windows 10 or higher / Ubuntu 21.04, GCC Compiler, IDE, Visual Studio

7. Course Plan:

Lecture No.	Topics	Recommended Books / Resources
1-6	**Arrays and Advanced Array Algorithms -1:** Frequency Arrays; Prefix Arrays; Two pointers and problems related to above mentioned topics: • Checking if two strings are anagrams. • Finding the mode of an array in constant time after pre-processing. • Finding the sum of elements within a given range. • Problems related to continuous subarrays, like finding maximum or minimum sums. • Finding all pairs that sum up to a given value. • Moving two pointers in opposite directions to find palindromes, meet in the middle, etc.	B03
7-10	**Arrays and Advanced Array Algorithms -2:** Advance Problems on Sorting: • Sorting an array to find the kth smallest or largest element. • Given an array with values representing colors (often {0, 1, 2} for three colors), sort them in place. • In a sorted array, find two numbers that add up to a specific target. This problem can be solved using two pointers after sorting.	B04
11-18	**Advanced String Algorithms:** Implementation Problems on Strings, Two pointers in Strings will be discussed and problems based on that. • Check if a given string is a palindrome by ignoring non-alphanumeric characters and case. • Given two strings, find the smallest substring in the first string that contains all characters of the second string. • Determine if a string can be constructed by repeating a substring. • Finding exact matches of short strings within larger texts. • Compress a string by replacing sequences of repeating characters with the character followed by the count (e.g., "aaabb" becomes "a3b2"). • Basic Calculator: Parse and evaluate a simple arithmetic expression. • String Matching Algorithms like KMP, Z Function, and Rabin Karp will be discussed and problems based on that.	B01
19-22	**Sliding Window:** Advance Problems on Sliding Window • Find the longest substring where all characters are unique using The sliding window approach • Counting elements that satisfy a condition within a window (e.g., number of distinct characters in a substring). • Given an array of integers and an integer K, find the maximum sum of a subarray of length K.	B02
23-24	**Recursion:** Basics of Recursion, basic problems on recursion to have better understanding of the call stack, Types of Recursion. Problems: • Given a sorted array, find the position of a target element using recursion. • Generate all permutations of a string. • Towers of Hanoi	B02
25-30	**Recursion Backtracking:** Implementation based Problems on recursion will be discussed in a thorough manner with call stacks, Introduction to Backtracking and problems on backtracking to get the flow of it:	B02

	• Given a set of distinct integers, generate all possible subsets (the power set). • Place n queens on an n x n chessboard so that no two queens threaten each other. • Given a grid where some cells are passable and others are blocked, find all paths for a rat to move from the top-left to the bottom-right cell. • Given an array of integers and a target, find all unique combinations where the sum equals the target.	
FA 1 (Syllabus Covered from Lecture 1 - 30 and Binary Search: Binary Search, Its Implementation and Advance Binary Search problems)		
31-34	**Advance problems on backtracking:** Recursion on Matrix: Discussion on how recursion and backtracking are used in general to solve problems on matrix/grid. • Given a 2D grid of letters and a word, determine if the word can be constructed from adjacent cells (horizontally or vertically). • In a grid with obstacles, find the number of unique paths from the top-left to the bottom-right corner. • Solve a 9x9 Sudoku board by filling empty cells so each row, column, and 3x3 grid contains unique digits from 1 to 9.	B02
35-38	**Linked List:** Interview Based Problems on LL. • Detect if a linked list has a cycle and remove it if one exists. • Given two sorted linked lists, merge them into a single sorted list. • Determine if a linked list is a palindrome. • Remove the N-th node from the end of a list in one pass.	B02
39-47	**Stacks**: Interview Problems Based on Stacks. • Given a string containing (,), {, }, [, and], determine if the string contains valid parentheses. • Design a stack that supports push, pop, top, and retrieving the minimum element in constant time. • Given an array representing the heights of bars in a histogram, find the area of the largest rectangle that can be formed. • Given a circular array of integers, find the next greater element for each element. If no such element exists, return -1.	B02
48-56	**Binary Trees and BST**: Standard Problems on Binary Trees and Binary Search Trees; Interview Problems Based on Trees. • Given the root of a binary tree, return the inorder traversal of its nodes' values. • Return the level order traversal of a binary tree's nodes. • Find the maximum depth of a binary tree. • Given a binary search tree (BST) and two nodes, find their lowest common ancestor (LCA). • Return all root-to-leaf paths in a binary tree. • Given preorder and inorder traversal of a tree, construct the binary tree. • Find the k-th smallest element in a BST.	B02, B05
ST1(Syllabus Covered from Lecture 1-56)		
57-67	**Hash maps** : Advance Hashing Techniques, Interview Problems based Hashing etc.; • Given a string and an integer K, find the length of the longest substring with at most K distinct characters. • Given two strings, s and t, find the smallest substring in s that contains all characters in t. • Given a string, find the length of the longest substring without repeating characters. • Given an array of integers and a target integer k, find the total number of subarrays whose sum equals k.	B01, B02, B05
68-75	**Priority Queue and Greedy Algorithms**: Advance Problems on Heaps. Implementation and Advance Problems on Greedy. • Find the kth largest element in an unsorted array.	B01, B02, B05

	• Given k sorted linked lists, merge them into one sorted list. • Given a non-empty array of integers, return the k most frequent elements. • Given a set of activities with start and finish times, select the maximum number of activities that don't overlap. • Given a set of characters and their frequencies, build the optimal prefix-free binary tree for data compression (Huffman Coding). •	
76-84	**Bit-masking**: Interview Based Problems on Bit Manipulations. • Given an array of integers, every element appears twice except for one. Find that single one. • Given an integer n, count how many 1 bits it has **(Brian Kernighan's Algorithm).** • Given an integer, reverse its bits. • Given an integer n, find how many times the number 1 appears in the binary representation of all numbers from 1 to n. • Given a set of distinct integers, return all possible subsets (the power set).	B04
	FA 2 **(Syllabus Covered from Lecture 57-84 and Concept of OOPS with real world examples)**	
85-92	**DP - 1**: DP company oriented questions and problems: • Given two strings, find the longest substring they share. This can be solved with dynamic programming or efficient rolling hashes. • Find the longest palindromic substring in a given string. This can be solved by expanding around centers or using dynamic programming. • Identify the longest substring that appears at least twice. Often solved with suffix arrays or suffix trees. • Using dynamic programming to find the edit distance between two strings.	B04
93-100	**DP – 2**: DP Patterns and Multi-dimensional DP • Match patterns with wildcards in strings. • Given a set of items, each with a weight and a value, determine the maximum value you can carry in a knapsack of limited capacity, where you can take fractions of items. • Given two strings, find the length of the longest subsequence that appears in both strings. • Given an array of integers, find the length of the longest strictly increasing subsequence. • Given a set of coin denominations, find the minimum number of coins required to make a given amount. • Given two strings, find the minimum number of operations (insertions, deletions, and substitutions) required to convert one string to another.	B02
101-104	**DP – 3**: DP on Trees. • Find the diameter of a binary tree. The diameter of a tree is the length of the longest path between any two nodes in the tree. • Find the maximum sum of values along any path from any node to any other node in a binary tree. The path can start and end at any node and doesn't necessarily pass through the root.	B02
	ST2(Syllabus Covered from Lecture 1-104)	
105-112	**Graphs and Tries**: Problems based on the graphs and Advance problems on Graphs. • Given an undirected or directed graph, represent it using an Using dynamic programming to find the edit distance between two strings. • Given a graph, use BFS to find the shortest path from a starting node to all other nodes. • Given a graph, use DFS to traverse all nodes. • Given a directed graph, detect if it contains a cycle. • Given a graph with non-negative weights, find the shortest path from a source	B02

	node to all other nodes using Dijkstra's algorithm. • Given a graph with weights (which could be negative), find the shortest path from the source node to all other nodes using the Bellman-Ford algorithm. • Given a directed acyclic graph (DAG), perform a **Topological Sort**. • Given a connected, undirected graph with weighted edges, find the Minimum Spanning Tree (MST) using **Kruskal's Algorithm**. • Basic problems, Range Queries and Interview Problems on Tries.	
END TERM – FULL SYLLABUS		

8. Delivery/Instructional Resources

Lecture Numbers	Topics	Web References	Audio-Video References
1-6	**Arrays and Advanced Array Algorithms -1:** Frequency Arrays; Prefix Arrays; Two pointers and problems related to above mentioned topics: • Checking if two strings are anagrams. • Finding the mode of an array in constant time after pre-processing. • Finding the sum of elements within a given range. • Problems related to continuous subarrays, like finding maximum or minimum sums. • Finding all pairs that sum up to a given value. • Moving two pointers in opposite directions to find palindromes, meet in the middle, etc.	https://www.geeksforgeeks.org/	https://www.youtube.com/watch?v=clKBWNdDE5c
7-10	**Arrays and Advanced Array Algorithms -2:** Advance Problems on Sorting: • Sorting an array to find the kth smallest or largest element. • Given an array with values representing colors (often {0, 1, 2} for three colors), sort them in place. • In a sorted array, find two numbers that add up to a specific target. This problem can be solved using two pointers after sorting.	https://www.geeksforgeeks.org/	https://www.youtube.com/watch?v=clKBWNdDE5c
11-18	**Advanced String Algorithms:** Implementation Problems on Strings, Two pointers in Strings will be discussed and problems based on that. • Check if a given string is a palindrome by ignoring non-alphanumeric characters and case. • Given two strings, find the smallest substring in the first string that contains all characters of the second string. • Determine if a string can be constructed by repeating a substring. • Finding exact matches of short strings within larger texts. • Compress a string by replacing sequences of repeating characters with the character followed by the count (e.g., "aaabb" becomes "a3b2"). • Basic Calculator: Parse and evaluate a simple arithmetic expression. • String Matching Algorithms like KMP, Z Function, and Rabin Karp will be discussed and problems based on that.	https://www.geeksforgeeks.org/	https://www.youtube.com/watch?v=clKBWNdDE5c

19-22	**Sliding Window:** Advance Problems on Sliding Window • Find the longest substring where all characters are unique using The sliding window approach • Counting elements that satisfy a condition within a window (e.g., number of distinct characters in a substring). • Given an array of integers and an integer K, find the maximum sum of a subarray of length K.	https://www.geeksforgeeks.org/	https://www.youtube.com/watch?v=clKBWNdDE5c
23-24	**Recursion:** Basics of Recursion, basic problems on recursion to have better understanding of the call stack, Types of Recursion. Problems: • Given a sorted array, find the position of a target element using recursion. • Generate all permutations of a string. • Towers of Hanoi	https://www.geeksforgeeks.org/	https://www.youtube.com/watch?v=clKBWNdDE5c
25-30	**Recursion Backtracking:** Implementation based Problems on recursion will be discussed in a thorough manner with call stacks, Introduction to Backtracking and problems on backtracking to get the flow of it: • Given a set of distinct integers, generate all possible subsets (the power set). • Place n queens on an n x n chessboard so that no two queens threaten each other. • Given a grid where some cells are passable and others are blocked, find all paths for a rat to move from the top-left to the bottom-right cell. • Given an array of integers and a target, find all unique combinations where the sum equals the target.	https://www.geeksforgeeks.org/recursive-functions/	https://www.youtube.com/watch?v=clKBWNdDE5c
31-34	**Advance problems on backtracking:** Recursion on Matrix: Discussion on how recursion and backtracking are used in general to solve problems on matrix/grid. • Given a 2D grid of letters and a word, determine if the word can be constructed from adjacent cells (horizontally or vertically). • In a grid with obstacles, find the number of unique paths from the top-left to the bottom-right corner. • Solve a 9x9 Sudoku board by filling empty cells so each row, column, and 3x3 grid contains unique digits from 1 to 9.	https://www.codingninjas.com/blog/2021/05/24/recursion-backtracking-algorithm-with-practice-problem/ https://www.geeksforgeeks.org/top-20-backtracking-algorithm-interview-questions/	https://www.youtube.com/watch?v=clKBWNdDE5c
35-38	**Linked List:** Interview Based Problems on LL. • Detect if a linked list has a cycle and remove it if one exists. • Given two sorted linked lists, merge them into a single sorted list. • Determine if a linked list is a palindrome. • Remove the N-th node from the end of a list in one pass.	https://www.geeksforgeeks.org/	https://www.youtube.com/watch?v=clKBWNdDE5c

39-47	**Stacks**: Interview Problems Based on Stacks. • Given a string containing (,), {, }, [, and], determine if the string contains valid parentheses. • Design a stack that supports push, pop, top, and retrieving the minimum element in constant time. • Given an array representing the heights of bars in a histogram, find the area of the largest rectangle that can be formed. • Given a circular array of integers, find the next greater element for each element. If no such element exists, return -1.	https://www.geeksforgeeks.org/	https://www.youtube.com/watch?v=clKBWNdDE5c
48-56	**Binary Trees and BST**: Standard Problems on Binary Trees and Binary Search Trees; Interview Problems Based on Trees. • Given the root of a binary tree, return the inorder traversal of its nodes' values. • Return the level order traversal of a binary tree's nodes. • Find the maximum depth of a binary tree. • Given a binary search tree (BST) and two nodes, find their lowest common ancestor (LCA). • Return all root-to-leaf paths in a binary tree. • Given preorder and inorder traversal of a tree, construct the binary tree. • Find the k-th smallest element in a BST.	https://www.geeksforgeeks.org/	https://www.youtube.com/watch?v=clKBWNdDE5c
57-65	**Hash maps** : Advance Hashing Techniques, Interview Problems based Hashing etc.; • Given a string and an integer K, find the length of the longest substring with at most K distinct characters. • Given two strings, s and t, find the smallest substring in s that contains all characters in t. • Given a string, find the length of the longest substring without repeating characters. • Given an array of integers and a target integer k, find the total number of subarrays whose sum equals k.	https://www.geeksforgeeks.org/	https://www.youtube.com/watch?v=clKBWNdDE5c
67-75	**Priority Queue and Greedy Algorithms**: Advance Problems on Heaps. Implementation and Advance Problems on Greedy. • Find the kth largest element in an unsorted array. • Given k sorted linked lists, merge them into one sorted list. • Given a non-empty array of integers, return the k most frequent elements. • Given a set of activities with start and finish times, select the maximum number of activities that don't overlap. • Given a set of characters and their frequencies, build the optimal prefix-free binary tree for data compression (Huffman Coding). •	https://www.geeksforgeeks.org/	https://www.youtube.com/watch?v=clKBWNdDE5c
76-84	**Bit-masking**: Interview Based Problems on Bit Manipulations. • Given an array of integers, every element appears twice except for one. Find that single one. • Given an integer n, count how many 1 bits it has **(Brian Kernighan's Algorithm).** • Given an integer, reverse its bits. • Given an integer n, find how many times the number 1	https://www.geeksforgeeks.org/	https://www.youtube.com/watch?v=clKBWNdDE5c

	appears in the binary representation of all numbers from 1 to n. • Given a set of distinct integers, return all possible subsets (the power set).		
85-92	**DP - 1**: DP company oriented questions and problems: • Given two strings, find the longest substring they share. This can be solved with dynamic programming or efficient rolling hashes. • Find the longest palindromic substring in a given string. This can be solved by expanding around centers or using dynamic programming. • Identify the longest substring that appears at least twice. Often solved with suffix arrays or suffix trees. • Using dynamic programming to find the edit distance between two strings.	https://www.geeksforgeeks.org/	https://www.youtube.com/watch?v=clKBWNdDE5c
93-100	**DP – 2**: DP Patterns and Multi-dimensional DP • Match patterns with wildcards in strings. • Given a set of items, each with a weight and a value, determine the maximum value you can carry in a knapsack of limited capacity, where you can take fractions of items. • Given two strings, find the length of the longest subsequence that appears in both strings. • Given an array of integers, find the length of the longest strictly increasing subsequence. • Given a set of coin denominations, find the minimum number of coins required to make a given amount. • Given two strings, find the minimum number of operations (insertions, deletions, and substitutions) required to convert one string to another.	https://www.geeksforgeeks.org/	https://www.youtube.com/watch?v=clKBWNdDE5c
101-104	**DP – 3**: DP on Trees. • Find the diameter of a binary tree. The diameter of a tree is the length of the longest path between any two nodes in the tree. • Find the maximum sum of values along any path from any node to any other node in a binary tree. The path can start and end at any node and doesn't necessarily pass through the root.	https://www.geeksforgeeks.org/	https://www.youtube.com/watch?v=clKBWNdDE5c
105-112	**Graphs and Tries**: Problems based on the graphs and Advance problems on Graphs. • Given an undirected or directed graph, represent it using an Using dynamic programming to find the edit distance between two strings. • Given a graph, use BFS to find the shortest path from a starting node to all other nodes. • Given a graph, use DFS to traverse all nodes. • Given a directed graph, detect if it contains a cycle. • Given a graph with non-negative weights, find the shortest path from a source node to all other nodes using Dijkstra's algorithm. • Given a graph with weights (which could be negative), find the shortest path from the source node to all	https://www.geeksforgeeks.org/detect-cycle-undirected-graph/ https://www.geeksforgeeks.org/topological-sorting/ https://practice.geeksforgeeks.org/problems/strongl	https://www.youtube.com/watch?v=clKBWNdDE5c

	other nodes using the Bellman-Ford algorithm. • Given a directed acyclic graph (DAG), perform a **Topological Sort**. • Given a connected, undirected graph with weighted edges, find the Minimum Spanning Tree (MST) using **Kruskal's Algorithm**. • Basic problems, Range Queries and Interview Problems on Tries	y-connected-components-kosarajus-algo/1 https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/ https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/ https://www.geeksforgeeks.org/bellman-ford-algorithm-dp-23/ https://www.codingninjas.com/codestudio/library/important- graph-problems-for-interviews-advanced-problems	

9. Lab Plan

Lab No.	Experiments	Learning Resources
1-6	**Arrays and Advanced Array Algorithms -1:** Frequency Arrays; Prefix Arrays; Two pointers and problems related to above mentioned topics: • Checking if two strings are anagrams. • Finding the mode of an array in constant time after pre-processing. • Finding the sum of elements within a given range. • Problems related to continuous subarrays, like finding maximum or minimum sums. • Finding all pairs that sum up to a given value. • Moving two pointers in opposite directions to find palindromes, meet in the middle, etc.	https://www.geeksforgeeks.org/
7-10	**Arrays and Advanced Array Algorithms -2:** Advance Problems on Sorting: • Sorting an array to find the kth smallest or largest element. • Given an array with values representing colors (often {0, 1, 2} for three colors), sort them in place. • In a sorted array, find two numbers that add up to a specific target. This problem can be solved using two pointers after sorting.	https://www.geeksforgeeks.org/
11-18	**Advanced String Algorithms:** Implementation Problems on Strings, Two pointers in Strings will be discussed and problems based on that.	

	• Check if a given string is a palindrome by ignoring non-alphanumeric characters and case. • Given two strings, find the smallest substring in the first string that contains all characters of the second string. • Determine if a string can be constructed by repeating a substring. • Finding exact matches of short strings within larger texts. • Compress a string by replacing sequences of repeating characters with the character followed by the count (e.g., "aaabb" becomes "a3b2"). • Basic Calculator: Parse and evaluate a simple arithmetic expression. • String Matching Algorithms like KMP, Z Function, and Rabin Karp will be discussed and problems based on that.	https://www.geeksforgeeks.org/
19-22	**Sliding Window:** Advance Problems on Sliding Window • Find the longest substring where all characters are unique using The sliding window approach • Counting elements that satisfy a condition within a window (e.g., number of distinct characters in a substring). • Given an array of integers and an integer K, find the maximum sum of a subarray of length K.	https://www.geeksforgeeks.org/
23-24	**Recursion:** Basics of Recursion, basic problems on recursion to have better understanding of the call stack, Types of Recursion. Problems: • Given a sorted array, find the position of a target element using recursion. • Generate all permutations of a string. • Towers of Hanoi	https://www.geeksforgeeks.org/
25-30	**Recursion Backtracking:** Implementation based Problems on recursion will be discussed in a thorough manner with call stacks, Introduction to Backtracking and problems on backtracking to get the flow of it: • Given a set of distinct integers, generate all possible subsets (the power set). • Place n queens on an n x n chessboard so that no two queens threaten each other. • Given a grid where some cells are passable and others are blocked, find all paths for a rat to move from the top-left to the bottom-right cell. • Given an array of integers and a target, find all unique combinations where the sum equals the target.	https://www.geeksforgeeks.org/recursive-functions/
31-34	**Advance problems on backtracking:** Recursion on Matrix: Discussion on how recursion and backtracking are used in general to solve problems on matrix/grid. • Given a 2D grid of letters and a word, determine if the word can be constructed from adjacent cells (horizontally or vertically). • In a grid with obstacles, find the number of unique paths from the top-left to the bottom-right corner. • Solve a 9x9 Sudoku board by filling empty cells so each row, column, and 3x3 grid contains unique digits from 1 to 9.	https://www.codingninjas.com/blog/2021/05/24/recursion- backtracking-algorithm-with-practice-problem/ https://www.geeksforgeeks.org/t op-20-backtracking-algorithm-interview-questions/
35-38	**Linked List:** Interview Based Problems on LL. • Detect if a linked list has a cycle and remove it if one exists. • Given two sorted linked lists, merge them into a single sorted list. • Determine if a linked list is a palindrome. • Remove the N-th node from the end of a list in one pass.	https://www.geeksforgeeks.org/
39-47	**Stacks**: Interview Problems Based on Stacks. • Given a string containing (,), {, }, [, and], determine if the string contains valid parentheses.	https://www.geeksforgeeks.org/

	• Design a stack that supports push, pop, top, and retrieving the minimum element in constant time. • Given an array representing the heights of bars in a histogram, find the area of the largest rectangle that can be formed. • Given a circular array of integers, find the next greater element for each element. If no such element exists, return -1.	
48-56	**Binary Trees and BST**: Standard Problems on Binary Trees and Binary Search Trees; Interview Problems Based on Trees. • Given the root of a binary tree, return the inorder traversal of its nodes' values. • Return the level order traversal of a binary tree's nodes. • Find the maximum depth of a binary tree. • Given a binary search tree (BST) and two nodes, find their lowest common ancestor (LCA). • Return all root-to-leaf paths in a binary tree. • Given preorder and inorder traversal of a tree, construct the binary tree. • Find the k-th smallest element in a BST.	https://www.geeksforgeeks.org/
57-65	**Hash maps** : Advance Hashing Techniques, Interview Problems based Hashing etc.; • Given a string and an integer K, find the length of the longest substring with at most K distinct characters. • Given two strings, s and t, find the smallest substring in s that contains all characters in t. • Given a string, find the length of the longest substring without repeating characters. • Given an array of integers and a target integer k, find the total number of subarrays whose sum equals k.	https://www.geeksforgeeks.org/
67-75	**Priority Queue and Greedy Algorithms**: Advance Problems on Heaps. Implementation and Advance Problems on Greedy. • Find the kth largest element in an unsorted array. • Given k sorted linked lists, merge them into one sorted list. • Given a non-empty array of integers, return the k most frequent elements. • Given a set of activities with start and finish times, select the maximum number of activities that don't overlap. • Given a set of characters and their frequencies, build the optimal prefix-free binary tree for data compression (Huffman Coding). •	https://www.geeksforgeeks.org/
76-84	**Bit-masking**: Interview Based Problems on Bit Manipulations. • Given an array of integers, every element appears twice except for one. Find that single one. • Given an integer n, count how many 1 bits it has **(Brian Kernighan's Algorithm).** • Given an integer, reverse its bits. • Given an integer n, find how many times the number 1 appears in the binary representation of all numbers from 1 to n. • Given a set of distinct integers, return all possible subsets (the power set).	https://www.geeksforgeeks.org/
85-92	**DP - 1**: DP company oriented questions and problems: • Given two strings, find the longest substring they share. This can be solved with dynamic programming or efficient rolling hashes. • Find the longest palindromic substring in a given string. This can be solved by expanding around centers or using dynamic programming. • Identify the longest substring that appears at least twice. Often solved with suffix arrays or suffix trees. • Using dynamic programming to find the edit distance between two strings.	https://www.geeksforgeeks.org/
93-100	**DP – 2**: DP Patterns and Multi-dimensional DP • Match patterns with wildcards in strings.	https://www.geeksforgeeks.org/

	• Given a set of items, each with a weight and a value, determine the maximum value you can carry in a knapsack of limited capacity, where you can take fractions of items. • Given two strings, find the length of the longest subsequence that appears in both strings. • Given an array of integers, find the length of the longest strictly increasing subsequence. • Given a set of coin denominations, find the minimum number of coins required to make a given amount. • Given two strings, find the minimum number of operations (insertions, deletions, and substitutions) required to convert one string to another.	
101-104	**DP – 3**: DP on Trees. • Find the diameter of a binary tree. The diameter of a tree is the length of the longest path between any two nodes in the tree. • Find the maximum sum of values along any path from any node to any other node in a binary tree. The path can start and end at any node and doesn't necessarily pass through the root.	https://www.geeksforgeeks.org/
105-112	**Graphs and Tries**: Problems based on the graphs and Advance problems on Graphs. • Given an undirected or directed graph, represent it using an Using dynamic programming to find the edit distance between two strings. • Given a graph, use BFS to find the shortest path from a starting node to all other nodes. • Given a graph, use DFS to traverse all nodes. • Given a directed graph, detect if it contains a cycle. • Given a graph with non-negative weights, find the shortest path from a source node to all other nodes using Dijkstra's algorithm. • Given a graph with weights (which could be negative), find the shortest path from the source node to all other nodes using the Bellman-Ford algorithm. • Given a directed acyclic graph (DAG), perform a **Topological Sort**. • Given a connected, undirected graph with weighted edges, find the Minimum Spanning Tree (MST) using **Kruskal's Algorithm**. • Basic problems, Range Queries and Interview Problems on Tries.	https://www.geeksforgeeks.org/detect-cycle-undirected-graph/ https://www.geeksforgeeks.org/topological-sorting/ https://practice.geeksforgeeks.org/problems/strongly-connected-components-kosarajus-algo/1 https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/ https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/ https://www.geeksforgeeks.org/bellman-ford-algorithm-dp-23/

		https://www.codingninjas.com/codestudio/library/important- graph-problems-for-interviews-advanced-problems
END TERM – FULL SYLLABUS**		

****Note: Topics and Problems mentioned above must be covered for the full syllabus. Additional problems related to topics of the syllabus above can be covered (if required).**

10. Action plan for different types of learners

Slow Learners	Average Learners	Fast Learners
• Remedial Classes on Saturdays • Encouragement for improvement using Peer Tutoring • Use of Audio and Visual Materials • Use of Real-Life Examples	• Workshops • Formative Exercises used to highlight concepts and notions • E-notes and E-exercises to read ahead of the pedagogic material.	• Engaging students to hold hands of slow learners by creating a Peer Tutoring Group • Design solutions for complex problems • Design solutions for complex problems • Presentation on topics beyond those covered in CHO

11. Evaluation Scheme & Components

Evaluation Component	Type of Component	No. of Assessments	Weightage of Component	Mode of Assessment
Component 1	Formative Assessment (FA)	02#	10%	Online(On-Campus)
Component 2	Sessional Tests (STs)	02*	30%	Online(On-Campus)
Component 3	End Term Examination	01	60%	Online(On-Campus)
Total		**100%**		

* Out of 02 STs, Best 1 ST for final marks evaluation of STs will be considered.
#Out of 02 FAs, Best 1 FA for final marks evaluation of FAs will be considered.
**As per Academic Guidelines, a minimum of 75% attendance is required to become eligible for appearing in the End Semester Examination

12. Syllabus of the Course:

Subject: Advanced Data Structures	**Subject Code:CS192**	
Topic(s)	**Lectures**	**Weightage**
Arrays and Advanced Array Algorithms -1: Frequency Arrays; Prefix Arrays; Two pointers and problems related to above mentioned topics: • Checking if two strings are anagrams. • Finding the mode of an array in constant time after pre-processing. • Finding the sum of elements within a given range. • Problems related to continuous subarrays, like finding maximum or minimum sums. • Finding all pairs that sum up to a given value. • Moving two pointers in opposite directions to find palindromes, meet in the middle, etc.	6	6%
Arrays and Advanced Array Algorithms -2: Advance Problems on Sorting: • Sorting an array to find the kth smallest or largest element. • Given an array with values representing colors (often {0, 1, 2} for three colors), sort them in place. • In a sorted array, find two numbers that add up to a specific target. This problem can be solved using two pointers after sorting.	4	6%
Advanced String Algorithms: Implementation Problems on Strings, Two pointers in Strings will be discussed and problems based on that. • Check if a given string is a palindrome by ignoring non-alphanumeric characters and case. • Given two strings, find the smallest substring in the first string that contains all characters of the second string. • Determine if a string can be constructed by repeating a substring. • Finding exact matches of short strings within larger texts. • Compress a string by replacing sequences of repeating characters with the character followed by the count (e.g., "aaabb" becomes "a3b2"). • Basic Calculator: Parse and evaluate a simple arithmetic expression.	8	6%
Sliding Window: Advance Problems on Sliding Window • Find the longest substring where all characters are unique using The sliding window approach • Counting elements that satisfy a condition within a window (e.g., number of distinct characters in a substring). • Given an array of integers and an integer K, find the maximum sum of a subarray of length K.	4	6%
Recursion: Basics of Recursion, basic problems on recursion to have better understanding of the call stack, Types of Recursion. Problems: • Given a sorted array, find the position of a target element using recursion. • Generate all permutations of a string. • Towers of Hanoi		

Recursion Backtracking: Implementation based Problems on recursion will be discussed in a thorough manner with call stacks, Introduction to Backtracking and problems on backtracking to get the flow of it: • Given a set of distinct integers, generate all possible subsets (the power set). • Place n queens on an n x n chessboard so that no two queens threaten each other. • Given a grid where some cells are passable and others are blocked, find all paths for a rat to move from the top-left to the bottom-right cell. • Given an array of integers and a target, find all unique combinations where the sum equals the target.		15%
Advance problems on backtracking: Recursion on Matrix: Discussion on how recursion and backtracking are used in general to solve problems on matrix/grid. • Given a 2D grid of letters and a word, determine if the word can be constructed from adjacent cells (horizontally or vertically). • In a grid with obstacles, find the number of unique paths from the top-left to the bottom-right corner. • Solve a 9x9 Sudoku board by filling empty cells so each row, column, and 3x3 grid contains unique digits from 1 to 9.	12	
Linked List: Interview Based Problems on LL. • Detect if a linked list has a cycle and remove it if one exists. • Given two sorted linked lists, merge them into a single sorted list. • Determine if a linked list is a palindrome. • Remove the N-th node from the end of a list in one pass.	4	6%
Stacks: Interview Problems Based on Stacks. • Given a string containing (,), {, }, [, and], determine if the string contains valid parentheses. • Design a stack that supports push, pop, top, and retrieving the minimum element in constant time. • Given an array representing the heights of bars in a histogram, find the area of the largest rectangle that can be formed. • Given a circular array of integers, find the next greater element for each element. If no such element exists, return -1.	8	6%
Binary Trees and BST: Standard Problems on Binary Trees and Binary Search Trees; Interview Problems Based on Trees. • Given the root of a binary tree, return the inorder traversal of its nodes' values. • Return the level order traversal of a binary tree's nodes. • Find the maximum depth of a binary tree. • Given a binary search tree (BST) and two nodes, find their lowest common ancestor (LCA). • Return all root-to-leaf paths in a binary tree. • Given preorder and inorder traversal of a tree, construct the binary tree. • Find the k-th smallest element in a BST.	8	10%
Hash maps : Advance Hashing Techniques, Interview Problems based Hashing etc.; • Given a string and an integer K, find the length of the longest substring with at most K distinct characters. • Given two strings, s and t, find the smallest substring in s that contains all characters in t. • Given a string, find the length of the longest substring without repeating characters. • Given an array of integers and a target integer k, find the total number of subarrays whose sum equals k.	8	6%

Priority Queue and Greedy Algorithms: Advance Problems on Heaps. Implementation and Advance Problems on Greedy. • Find the kth largest element in an unsorted array. • Given k sorted linked lists, merge them into one sorted list. • Given a non-empty array of integers, return the k most frequent elements. • Given a set of activities with start and finish times, select the maximum number of activities that don't overlap. • Given a set of characters and their frequencies, build the optimal prefix-free binary tree for data compression (Huffman Coding)		10%
Bit-masking: Interview Based Problems on Bit Manipulations. • Given an array of integers, every element appears twice except for one. Find that single one. • Given an integer n, count how many 1 bits it has **(Brian Kernighan's Algorithm).** • Given an integer, reverse its bits. • Given an integer n, find how many times the number 1 appears in the binary representation of all numbers from 1 to n. • Given a set of distinct integers, return all possible subsets (the power set).	15	
DP - 1: DP company oriented questions and problems: • Given two strings, find the longest substring they share. This can be solved with dynamic programming or efficient rolling hashes. • Find the longest palindromic substring in a given string. This can be solved by expanding around centers or using dynamic programming. • Identify the longest substring that appears at least twice. Often solved with suffix arrays or suffix trees. • Using dynamic programming to find the edit distance between two strings.		18%
DP – 2: DP Patterns and Multi-dimensional DP • Match patterns with wildcards in strings. • Given a set of items, each with a weight and a value, determine the maximum value you can carry in a knapsack of limited capacity, where you can take fractions of items. • Given two strings, find the length of the longest subsequence that appears in both strings. • Given an array of integers, find the length of the longest strictly increasing subsequence. • Given a set of coin denominations, find the minimum number of coins required to make a given amount. • Given two strings, find the minimum number of operations (insertions, deletions, and substitutions) required to convert one string to another.		
DP – 3: DP on Trees. • Find the diameter of a binary tree. The diameter of a tree is the length of the longest path between any two nodes in the tree. • Find the maximum sum of values along any path from any node to any other node in a binary tree. The path can start and end at any node and doesn't necessarily pass through the root.	24	
Graphs and Tries: Problems based on the graphs and Advance problems on Graphs. • Given an undirected or directed graph, represent it using an Using dynamic programming to find the edit distance between two strings. • Given a graph, use BFS to find the shortest path from a starting node to all other nodes. • Given a graph, use DFS to traverse all nodes. • Given a directed graph, detect if it contains a cycle. • Given a graph with non-negative weights, find the shortest path from a source node to all other nodes using Dijkstra's algorithm. • Given a graph with weights (which could be negative), find the shortest path from	11	5%

the source node to all other nodes using the Bellman-Ford algorithm. • Given a directed acyclic graph (DAG), perform a **Topological Sort**. • Given a connected, undirected graph with weighted edges, find the Minimum Spanning Tree (MST) using **Kruskal's Algorithm**. • Basic problems, Range Queries and Interview Problems on Tries.		

This Document is approved by:

Designation	**Name**	**Signature**
Course Coordinator	Dr. Suhasini	
Head-Academic Delivery	Dr. Susheela Hooda	
Dean	Dr. Rupali Gill	
Date	10.01.2025	