Lesson Plan for DSA Sessions: BTech 2nd Year Even Sem 2024-25.

Course: Data Structures and Algorithms (DSA)
Program: B.Tech 2nd Year (CSE - All Branches)
Semester: Even Semester (2024-25)
Session: TBPPP - DSA

Introduction:

This lesson plan is designed to provide a structured and comprehensive approach to teaching Data Structures and Algorithms (DSA) under the **Target-Based Placement Preparation Program (TBPPP)**. The plan is aligned with the academic objectives of the Even Semester for the 2024-25 session and aims to prepare students with strong problem-solving and coding skills essential for technical placements.

Course Structure:

The course is divided into three integrated components to ensure holistic learning:

1. **Theory Sessions (3 hrs or 4 hrs per week):**
   Focused on conceptual understanding, these sessions cover key DSA topics in depth, following a week-wise progression to build a strong foundation.
2. **Custom Problem (Lab 2 hrs per week):**
   Hands-on practice of carefully designed coding problems to help students apply theoretical concepts and develop practical problem-solving abilities.
3. **LeetCode Practice (PCPH - 2 hrs per week):**
   Dedicated practice sessions to solve curated LeetCode problems aimed at reinforcing concepts and enhancing coding proficiency.

13-week DSA Plan with Custom Problems and **LeetCode problem names and links**:

Week	Topics	Theory(3/4 Hours)	Practice Problems (2 Hours Lab)	Leet Code (2 Hours Cross Section) PCPH
1	Introduction to DSA and Arrays	Importance of DSA, Complexity Analysis, Arrays Basics	- Initialize arrays, take input and print arrays - Find max/min in array - Reverse array - Rotate array by k steps DIY: - Count frequency of each element - Check if array is sorted - Find second largest element	Pascal Triangle, Buy and Sell Stock, Move Zeroes
2	Array Operations	Insertion, Deletion, Linear Search, Binary Search, Sorting (Bubble,	- Search for an element using Linear Search and Binary Search - Find the first and last occurrence of a number in sorted array	3Sum, Remove Duplicates from Sorted Array, Container with Most Water,

Week	Topics	Theory(3/4 Hours)	Practice Problems (2 Hours Lab)	Leet Code (2 Hours Cross Section) PCPH
		Insertion, Selection Sort)	DIY: - Sort array using Bubble, Selection, and Insertion Sort - Rotate matrix 90 degrees clockwise	
3	Strings	String operations: Concatenation, Substring Search, Palindrome Check, Detecting Anagrams String Builder & Buffer	- Check if a string is palindrome - Reverse string and words in a string - Count vowels and consonants DIY: - Detect anagrams - Find the longest common prefix - Count the number of substrings with a specific character	Valid Palindrome, Reverse String, Reverse Vowels of a String, First Unique Character
4	Recursion and Backtracking	Recursive Functions, Quick Sort, Merge Sort, Backtracking Template, N-Queens	- Solve factorial using recursion - Print Fibonacci, Coin Change, Generate Paranthesis, Permutations,Keypad Combinations DIY: - Solve Tower of Hanoi - Solve Climbing Stairs - Count subsequences Unique Permutations matching a given condition	Climbing Stairs, Tower of Hanoi, Merge Sort,N-Queens,
5	Singly Linked List	Implementation, Operations (Insert, Delete, Traverse), Detect and Remove Cycle	- Implement linked list operations: AddFirst, AddLast, DeleteAt, Search Target DIY: - Reverse linked list iteratively and recursively - Find intersection point of two linked lists - Detect and remove cycles	Intersection of Two Linked Lists, Find Middle of Linked List, Linked List Palindrome
6	Doubly & Circular Linked Lists	Implementation and Operations, Comparison with Singly Linked List	- Implement doubly linked list operations - Traverse circular linked list - Delete nodes at specific positions DIY: Search an element in a particular position in a	Reverse Linked List II, Rotate List

Week	Topics	Theory(3/4 Hours)	Practice Problems (2 Hours Lab)	Leet Code (2 Hours Cross Section) PCPH
			doubly linked list, reversing a doubly linked list and rotating a circular list	
7	Stacks	Stack operations: Push, Pop, Peek, Applications	- Implement stack using arrays and linked lists - Evaluate postfix expressions Rain water , Histogram Reverse Stack using Recursion	Valid Parentheses, Next Greater Element, Min Stack
			DIY: - Implement two stacks in one array - Check for balanced parentheses - Find span of stock prices	
8	Queues	Queue operations: Enqueue, Dequeue, Circular Queue, Priority Queue	- Implement queue using arrays and linked lists - Implement circular queue	Implement Queue using Stacks, Sliding Window Maximum
			DIY: - Design a priority queue - Find the first negative integer in every window of size k	
9	Trees (Binary Trees)	Preorder, Inorder, Postorder Traversals	- Implement tree traversal algorithms - Count number of leaf nodes - Check if a binary tree is balanced, Hight of BT , Path sun I,II,III	Symmetric Tree, Maximum Depth of Binary Tree, Diameter of Binary Tree
			DIY - Find the height of a binary tree - Print all paths from root to leaf	
10	Binary Search Trees (BST)	BST Properties, Operations (Search, Insert, Delete), Applications	- Implement BST operations: Insert, Delete, Search - Find the lowest common ancestor	Validate BST, Convert Sorted Array to BST
			DIY: - Find kth smallest and largest elements - Check if a tree is a valid BST	

Week	Topics	Theory(3/4 Hours)	Practice Problems (2 Hours Lab)	Leet Code (2 Hours Cross Section) PCPH
11	Graphs: BFS & DFS	Graph Representation (Adjacency Matrix/List), BFS, DFS	- Implement BFS and DFS algorithms - Detect a cycle in a graph - Find connected components	Number of Islands, Clone Graph
12	Graph Algorithms	Dijkstra's Algorithm, Minimum Spanning Tree (Prim's, Kruskal's)	- Implement Dijkstra's algorithm - Implement Prim's and Kruskal's MST algorithms - Solve shortest path problems	Network Delay Time, Minimum Cost to Connect All Points
13	Advanced Topics & Revision	DP, Hashing, Sliding Window, Bit Manipulation, Revision of All Topics	- Solve Fibonacci using DP - Count number of ways to reach the nth step - Solve subset sum problem DIY: - Solve subarray problems using sliding window - Count distinct elements in every window - Use bit manipulation for XOR tricks	Sliding Window Maximum, Subarray Sum Equals K

Objectives:

- To ensure conceptual clarity and in-depth understanding of fundamental and advanced DSA topics.
- To provide ample hands-on coding practice, preparing students for technical interviews and coding assessments.
- To create a seamless learning experience by aligning theoretical knowledge with practical problem-solving.

Guidelines:

- Faculty members are advised to follow the week-wise schedule provided in the lesson plan to maintain uniformity across all sections.
- Coding problems and LeetCode links have been specified for each topic to streamline lab and PCPH sessions.
- Faculty should encourage active participation and provide support to students in solving problems during lab and PCPH sessions.

This lesson plan is a roadmap for delivering a consistent and effective learning experience, ensuring students are well-prepared for academic and placement challenges. Let us work together to achieve the best outcomes for our students!