AD-1 ENDSEM ODD-2023

L 01	Introduction to the course/subject: Program Outcomes; Course Outcomes; Lesson plan; Teaching methodology; Evaluation strategy etc.
L 02	**Introduction to Algorithm Design:** Importance of problem solving using algorithms; Characteristic features of an algorithm(input, output, finiteness, definiteness, effectiveness, correctness, efficiency);
L 03	**Introduction to Algorithm Design:** Expressing algorithms (pseudocode); Basic aspects of algorithms (correctness, design and analysis)
L 04	**Computational tractability:** Polynomial time as a definition efficiency of an algorithm; Worst case Running times and Brute-Force Search
L 05	**Asymptotic order of growth** (Big-Oh, Big-Omega, Big-Theta)
L 06	**Asymptotic order of growth** (Big-Oh, Big-Omega, Big-Theta)
L 07	**Recurrences** (Iterative, Substitution and Master method)
L 08	**Recurrences** (contd..)
L 09	**Priority Queue Implementation using Heap data structure**
L 10	**Priority Queue Implementation using Heap data structure**
L 11	**Graph:** Basic definitions, applications and representations
L 12	**Graph:** Basic definitions, applications and representations (contd..)
L 13	**Graph:** Graph connectivity and graph traversal (BFS, DFS)
L 14	**Graph:** Graph connectivity and graph traversal (BFS, DFS)
L 15	**Graph:** Testing bipartiteness – an application of BFS
L 16	**Graph:** Connectivity in directed graph; Directed-Acyclic-Graph and Topological ordering
L 17	**Graph:** Connectivity in directed graph; Directed-Acyclic-Graph and Topological ordering
L 18	**Graph:** MST using Kruskal's algorithm—the union-find data structure
L 19	**Graph:** MST using Kruskal's algorithm—the union-find data structure (contd..)

L 20	**Graph:** MST using Prim's algorithm
L 21	**Graph:** Shortest path problem (Dijkstra' algorithm)
L 22	**Greedy Method:** Interval Scheduling with proof of optimality using the Greedy Algorithm Stays Ahead
L 23	**Greedy Method:** Interval Scheduling with proof of optimality using the Greedy Algorithm Stays Ahead
L 24	**Greedy Method:** Scheduling to Minimize Lateness with proof of optimality using An Exchange Argument
L 25	**Greedy Method:** Optimal Caching: A More Complex Exchange Argument (no discussion on proof of optimality)
L 26	**Greedy Method:** Huffman Codes and Data Compression (no discussion on proof of optimality)
L 27	**Greedy Method:** Huffman Codes and Data Compression (no discussion on proof of optimality) contd..
L 28	**Divide and Conquer:** Control abstraction
L 29	**Divide and Conquer:** Merge sort
L 30	**Divide and Conquer:** Counting inversions
L 31	**Divide and Conquer:** Quick sort
L 32	**Divide and Conquer:** Quick sort
L 33	**Divide and Conquer:** Fast integer multiplication (Karatsuba algorithm)
L 34	**Dynamic Programming:** Principles of Dynamic Programming (Memoization or Iteration over Subproblems)
L 35	**Dynamic Programming:** Weighted Interval Scheduling
L 36	**Dynamic Programming:** Subset Sums and Knapsacks
L 37	**Dynamic Programming:** RNA Secondary Structure
L 38	**Dynamic Programming:** Sequence Alligment
L 39	**Dynamic Programming:** Shortest Paths in a Graph (Bellman-Ford algorithm); Negative-Weight-Cycles

LABS:- LAB1,LAB2,LAB3,
LAB4-SORTING,LAB-5:SORTING,LAB6-SEARCHING.
LAB7-SEARCHING,LAB8-LINKED LIST,
LAB-9:JAVA COLLECTIONS,LAB10-STACK,
LAB11-QUEUE,LAB12-TREE,LAB13-PRIORITYQUEUE,
LAB14-GRAPHS,LAB15-HASH TABLE