MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

COURSE PLAN

Department	**:**	Department of Data Science and Computer Applications
Course Name & code	**:**	Design and Analysis of Algorithms Lab & DSE-2264
Semester & branch	**:**	IV Semester B. Tech DSE
Name of the faculty	**:**	Linda Varghese & Dr. Sandhya Dubey
No of contact hours/week:		

L	**T**	**P**	**C**
0	0	3	1

COURSE OUTCOMES (COs)

	At the end of this course, the student should be able to:	**No. of Contact Hours**	**Marks**
CO1:	Ability to design one or more algorithms for a problem using appropriate data structures	12	
CO2:	Convert the algorithm into a program which is efficient	12	
CO3:	Ability to determine the complexity of various algorithms or the resulting program	12	
	Total		

ASSESSMENT PLAN

1. Continuous Evaluation	60%
Total of 3 regular evaluations which will be carried out in alternate weeks. Each evaluation is for 20 marks of which will have the following split up: Record: 4 Marks; Viva/Test/Quiz: 12 Marks; Execution: 4 Marks; Total = 20 Marks Total Internal Marks: 3 * 20 =60 Marks	
2. Lab Examination	40%
• Examination of 2 hours duration (Max. Marks: 40). Program Write up: 15 Marks; Program Execution: 25; Marks Total: 15+25 =40 Marks	

DETAILED COURSE PLAN

L. No.	Topics	Course Outcomes Addressed
	LAB- 1 **L1.1)** Write an algorithm and program to implement a doubly linked list which supports the following operations i. Create the list by adding each node at the front ii. Insert a new node to the left of the node whose key value is read as an input. iii. Delete all occurrences of a given key, if it is found. Otherwise, display appropriate message iv. Search a node based on its key value v. Display the contents of the list **L1.2)**. Write a program to construct a binary tree to support the following operations. Assume no duplicate elements while constructing the tree. i. Given a key, perform a search in the binary tree. If the key is found then display "key found" else inserts the key in the BST. ii. Display the tree using inorder, preorder and post order traversal methods **L1.3)**. Write a program to implement the following graph representations. i. Adjacency list ii. Adjacency matrix **Additional Questions: -** **L1. a)** Repeat problem given in solved exercise using singly linked list. **L1. b) W**rite a program to implement Stack and Queue using circular doubly linked list.	CO1, CO2, CO3

LAB- 2

L2. 1) Write an algorithm for finding the Greatest Common Divisor (GCD) of two numbers using Euclid's algorithm. Write a program to implement the Euclid's algorithm.

L2. 2) Write a program to find GCD using consecutive integer checking method and analyze its time efficiency.

L2. 3) Write a program to find GCD using middle school method and analyze its time efficiency.

Additional Questions:

L2. a) Write a program for computing $\lfloor \sqrt{n} \rfloor$ for any positive integer and analyze its time efficiency. Besides assignment and comparison, your program may only use the four basic arithmetic operations.

L2. b) Write a program to implement recursive solution to the Tower of Hanoi puzzle and analyze its time efficiency.

L2. c) Write a program to compute the $\mathbf{n^{th}}$ Fibonacci number recursively and analyze its time efficiency.

LAB-3

L3. 1) Write a program to sort a set of integers using selection sort algorithm and analyze its time efficiency.

L3. 2) Write a program to sort set of integers using bubble sort. Analyze its time efficiency.

L3. 3) Write a program to implement brute-force string matching. Analyze its time efficiency.

L3. 4) Write a program to implement matrix multiplication using brute-force technique and analyze its time efficiency.

Additional Questions:

L3. a) Write a program to implement solution to partition problem using brute-force technique and analyze its time efficiency. A partition problem takes a set of numbers and finds two disjoint sets such that the sum of the elements in the first set is equal to the second set.

L3. b) Write a program in C for finding maximal clique in a graph by brute-force approach. Clique is a maximal complete subgraph in a graph.

LAB-4

CO1, CO2, CO3

L4. 1) Write a program to implement Knapsack problem using brute-force design technique and analyze its time efficiency.
Knapsack Problem: Given *n* items of known weights w_1, w_2 ... w_n values v_1, v_2, ... v_n and a knapsack of capacity *B*, find the most valuable subset of items that fit into the knapsack.

L4. 2) Write a program for assignment problem by brute-force technique. Analyze its time efficiency.

L4. 3) Write a program for depth-first search of a graph. Identify the push and pop order of vertices.

Additional Questions:

L4. a) Write a program for breadth-first search of a graph.

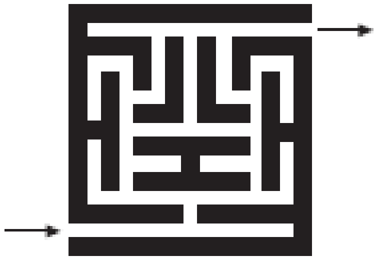
L4. b) Write a program to check whether a graph is bipartite or not using:

i) DFS to check for bipartite

ii) BFS to check for bipartite

Note: A graph is said to be bipartite if all its vertices can be partitioned into two disjoint subsets X and Y so that every edge connects a vertex in X with a vertex in Y.

L4. c) Write a program to construct a graph for the following maze. One can model a maze by having a vertex for a starting point, a finishing point, dead ends, and all the points in the maze where more than one path can be taken, and then connecting the vertices according to the paths in the maze. Also find the solution to the maze using Depth-First-Search.



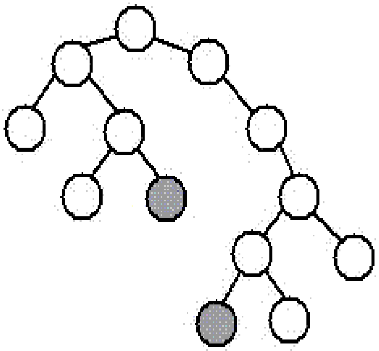
LAB-5

L5. 1) Write a program to sort a set of numbers using insertion sort and analyze its time efficiency.

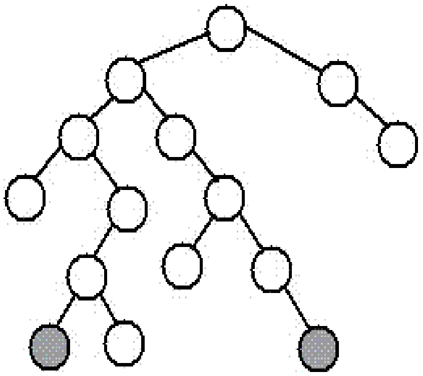
L5. 2) Write a program to determine the Topological sort of a given graph using:

- **i)** Depth-First technique
- **ii)** Source removal technique

L5. 3) Write a program to find diameter of a binary tree. Diameter of a binary tree is the longest path between any two nodes.
For e.g. consider the following two binary trees.



diameter, 9 nodes, through root



diameter, 9 nodes, NOT through root

The diameter in each tree is 9. In the first one, longest path passes through root whereas in the second it does not.

Additional Questions:

L5. a) A student management system should be developed by a software company. There are certain dependent and independent modules that must be developed by K teams. The unit testing and integration testing is done before the deployment of the product. Design an algorithm using decrease and conquer technique such that it gives a schedule of tasks in the order in which it must be executed.

L5. b) Modify the algorithm such that it checks whether there is any task dependency between the teams.

CO1, CO2, CO3

	L5. c) Write a program in C to generate Fibonacci numbers upto n where n is a positive integer. **L5. d)** Write a program in C to reverse a queue.	
	<u>LAB-6</u> **L6. 1)** Write a program to sort given set of integers using Quick sort and analyze its efficiency. **L6. 2)** Write a program to sort given set of integers using Merge sort and analyze its efficiency. **L6. 3)** Write a program to determine the height of a binary search tree and analyze its time efficiency. **L6. 4)** Find total number of nodes in a binary tree and analyze its efficiency. **Additional Questions:** **L6. a)** Write a program to find an where n > 0 using divide and conquer strategy. **L6. b)** Write a program to implement binary-search using divide and conquer strategy. **L6. c)** Write a program to find the total number of nodes in a binary tree and analyze its efficiency. Obtain the experimental result of order of growth and plot the result.	CO1, CO2, CO3
	<u>LAB-7</u> **L7. 1)** Write a program to create a binary search tree and display its elements using all the traversal methods and analyse its time efficiency. **L7. 2)** Write a program to create a AVL tree for a set of integers and analyze its time efficiency. For the AVL tree created, insert an element 6.	CO1, CO2, CO3

	Additional Questions: **L7. a)** For the AVL tree created in exercise 1 above, insert an element 6	
	LAB-8 **L8. 1)** Write a program to sort the list of integers using heap sort with bottom-up max heap construction and analyze its time efficiency. Prove experimentally that the worst-case time complexity is O (n log n) **L8. 2)** Write a program to create a heap for the list of integers using top-down heap construction algorithm and analyse its time efficiency. **L8. 3)** Write a program to sort the list of integers using heap sort technique and analyse its time efficiency. **L8. 4)** Write a program for finding and deleting an element of the smallest value in a heap. **Additional Questions:** **L8. a)** Write a program to check whether an array H[1..n] is a heap or not **L8. b)** Write a program for finding and deleting an element of a given value in a heap. **LAB-9** **L9. 1)** Write a program to sort set of integers using comparison counting algorithm. **L9. 2)** Write a program to implement Horspool's algorithm for String Matching and find the number of key comparisons in successful search and unsuccessful search. **L9. 3)** Write a program to construct the Open hash table. Find the number of key comparisons in successful search and unsuccessful search. **Additional Questions:** **L9. a)** Write a program to construct the closed hash table. Find the number of key comparisons in successful search and unsuccessful search.	CO1, CO2

	L9. b) Write a program to implement Boyer-Moore algorithm for String Matching and find the number of key comparisons in successful search and unsuccessful search. **L9. c)** Write a program to sort the elements using distribution counting method.	
	<u>LAB-10</u> **L10. 1)** Write a program to find the Binomial Co-efficient using Dynamic Programming. **L10. 2)** Write a program to compute the transitive closure of a given directed graph using Warshall's algorithm and analyse its time efficiency. **L10. 3)** Write a program to implement Floyd’s algorithm for the All-Pairs-Shortest-Paths problem for any given graph and analyse its time efficiency. **L10. 4)** Write a program to implement 0/1Knapsack problem using bottom-up dynamic programming. **Additional Questions:** **L10. a)** Write a program to implement **0/1** knapsack problem using memory functions. **L10. b)** Write a program that finds the composition of an optimal subset from the table generated by the bottom-up dynamic programming algorithm for the knapsack problem.	CO1, CO2, CO3

	LAB-11 **L11. 1)** Write a program to find Minimum Cost Spanning Tree of a given undirected graph using Prim’s algorithm. **L11. 2)** Write a program to find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm and analyse its time efficiency. **L11. 3)** Write a program to find shortest path from a given vertex to other vertices in a given weighted connected graph, Using Dijkstra’s algorithm and analyse its time efficiency. **Additional Questions:** **L11. a)** Write a program to implement Huffman tree construction algorithm. **L11. b)** Write a program to find a maximum spanning tree – a spanning tree with the largest possible edge weight of a weighted connected graph. **L11. c)** Write a program to implement the greedy algorithm for the change-making problem, with an amount n and coin denominations $\mathbf{d_1 > d_2 > ... > d_m}$ as its input.	CO1, CO2, CO3
	LAB-12 **L12. 1)** Write a program for N-queens problem using backtracking technique. **L12. 2)** Write a program to find the solution to the subset-sum problem for S= {1, 2, 5, 6, 8} and d=9, using backtracking. **L12. 3)** Write a program to implement Knapsack problem using branch and bound technique. **Additional Questions:**	CO1, CO2

	L12. a) Write a program for finding Hamiltonian circuit for the graph, using backtracking. **L12. b)** Write a program to implement assignment problem using Branch and Bound.	

References:

1. Anany Levitin, Introduction to The Design and Analysis of Algorithms, 3rd Edition, Pearson Education, India, 2012.
2. Ellis Horowitz, SartajSahni, Dinesh Mehta, Fundamentals of Data Structures in C++, 2nd Edition, GalgotiaPublications, Reprint 2013
3. Thomas H. Cormen, Charles E. Leiserson, Ronal L, Rivest, Clifford Stein, Introduction to Algorithms, Phi, 2nd Edition, 2006.

Submitted by: Linda varghese & Dr.Sandhya Dubey

(Signature of the faculty)

Date:

Approved by:

(Signature of HOD)

Date: 30-02-2023

FACULTY MEMBERS TEACHING THE COURSE (IF MULTIPLE SECTIONS EXIST):

FACULTY	SECTION	FACULTY	SECTION
Linda varghese	**B**		
Dr.Sandhya Dubey	A		

****