NIRMA UNIVERSITY

Institute:	Institute of Technology, School of Technology
Name of Programme:	MTech CSE, MTech CSE (Cyber Security), MTech CSE (Data Science)
Course Code:	6CS402CC22
Course Title:	Data Structures and Algorithms
Course Type:	Core
Year of Introduction:	2022-23

L	T	Practical Component				C
		LPW	PW	W	S	
3	0	2	-	-	-	4

Course Learning Outcomes (CLO):

At the end of the course, the student will be able to –

1. relate to time & space complexity and formal aspects of algorithms (BL2)
2. identify appropriate data structures and approaches for efficient algorithm design (BL3)
3. analyse algorithmic techniques to solve computational problems (BL3)
4. design and implement efficient algorithms using various approaches. (BL6)

Unit	Contents	Teaching Hours (Total 45)
Unit-I	**Introduction to Data Structure and Algorithms:** Notions of time and space complexity, Asymptotic Analysis, Recurrence relation, Overview of linear and nonlinear data structures such as Linked List, Tree, Binary Heap, Binomial Heap, Fibonacci Heap	12
Unit-II	**Divide and Conquer**: General template, Various algorithm implementations like Binary search, Merge sort, Quick sort, etc.	09
Unit-III	**Greedy Algorithms:** Making change, graphs and minimum spanning tree, shortest path, Knapsack problem, Scheduling, etc.	09
Unit-IV	**Dynamic Programming**: Introduction of Dynamic Programming, Principle of Optimality, Examples like Single source shortest paths, Knapsack problem, Chained matrix multiplication, Longest common subsequence, etc.	08
Unit-V	**Graph Algorithms**: Elementary algorithms, Depth-first search, Breadth First Search, Disjoint set structures, Backtracking, and Branch & Bound techniques with related examples.	07

Self-Study:

The self-study contents will be declared at the commencement of the semester. Around 10% of the questions will be asked from self-study content.

Suggested Readings/ References:

1. Gilles Brassard and Paul Bratley, Fundamentals of Algorithmics, Prentice Hall
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest& Clifford Stein, Introduction

to Algorithms, Prentice Hall
3. Ellis Horowitz, SartajSahni, Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms, University Press
4. Jean-Paul Tremblay and Paul G. Sorenson, An Introduction to Data Structures with Applications, McGraw Hill
5. Robert L. Kruse, Data Structures and Program Design in C, Prentice Hall.

Suggested List of Experiments:

Sr. No.	Name of Experiments/Exercises	Hours
1	Implement iterative and full recursive versions of the following sorting techniques and repeat the experiment for larger values of n with different input patterns, and **plot the comparison graph of a number of elements versus execution time taken**. The elements can be read from a file or can be generated using a random number generator. Selection sort Bubble sort Insertion sort	02
2	a. Implement linear and binary searching and compare the time complexity of both algorithms for larger data sets. b. You are given a sorted array consisting of only integers where every element appears exactly twice, except for one element, which appears exactly once. Return the single element that appears only once. Your solution must run in O(log n) time and O(1) space. **Perform on leet code platform**. https://leetcode.com/problems/single-element-in-a-sorted-array/	02
3	a. Implement recursive algorithm for the following and evaluate the time complexity of each algorithm. -Finding Factorial -Sum of n numbers -Tower of Hanoi -Finding min-max b. Given an integer n, return true if it is a power of four. Otherwise, return false. An integer n is a power of four, if there exists an integer x such that n == 4x. **Perform this on leet code platform**. https://leetcode.com/problems/power-of-four/	04
4	**Solve the following problems on Leet Code coding platform** a. Given the head of a singly linked list where elements are sorted in ascending order, convert sorted list to a binary search tree. https://leetcode.com/problems/convert-sorted-list-to-binary-search-tree/ b. Given the head of a singly linked list, return the middle node of the linked list. If there are two middle nodes, return the second middle node https://leetcode.com/problems/middle-of-the-linked-list/	04

5	a.	Solve the Make-Change problem using the Greedy approach and dynamic programming.	04
	b.	Given a string s, return true if the s can be palindrome after deleting at most one character from it. **Perform on leet code platform** https://leetcode.com/problems/valid-palindrome-ii/	
	c.	You are given an integer array coins representing coins of different denominations and an integer amount representing a total amount of money. Return the number of combinations that make up that amount. If that amount of money cannot be made up by any combination of the coins, return 0. **Perform on leet code platform** https://leetcode.com/problems/coin-change-ii/description/	
6		Implement Kruskal's algorithm to find MST using a greedy approach.	04
7		Implement Job Scheduling problems using a greedy approach.	02
8		Implement 0-1 knapsack problem using dynamic programming.	02
9		Given two sequences X and Y, find the longest common subsequence (LCS) of X and Y using dynamic programming.	02
10		Given a list of words, a list of single letters (might be repeating), and a score of every character. Return the maximum score of any valid set of words formed by using the given letters (words[i] cannot be used two or more times). It is not necessary to use all characters in letters, and each letter can only be used once. Score of letters 'a', 'b', 'c', ... ,'z' is given by score[0], score[1], ... , score[25] respectively. https://leetcode.com/problems/maximum-score-words-formed-by-letters/description/	04