NIRMA UNIVERSITY

Institute:	Institute of International Study
Name of Programme:	Bachelor of Science (Computer Science and Engineering) [2+2 Dual Degree]
Faculty	Faculty of Technology & Engineering
Course Code:	2CS508
Course Title:	Introduction to Design and Analysis of Algorithms
Course Type:	Core
Year of Introduction:	2023-24

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

Course Learning Outcomes (CLO):

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

1. explain the logic of fundamental algorithms and comprehend the notion of algorithmic complexity. (BL2)
2. apply algorithms techniques in real-life problem-solving. (BL3)
3. apply concepts of data structures and greedy algorithms. (BL3)
4. evaluate suitable data structures to solve a problem effectively and efficiently. (BL5)

Syllabus: **Total Teaching hours: 30**

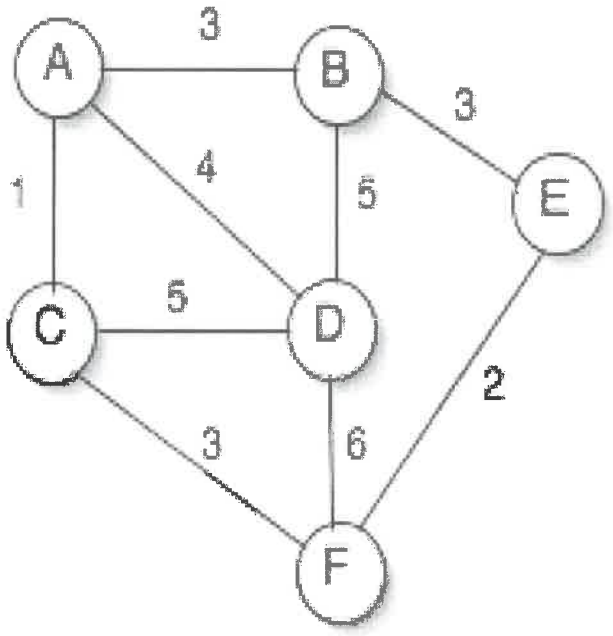
Unit	Syllabus	Teaching hours
Unit-I	**Elementary Algorithmic:** Efficiency of Algorithms, Average and worst-case analysis, Elementary Operation	02
Unit-II	**Analysis Techniques:** Empirical, mathematical, Asymptotic analysis and related unconditional and conditional notations **Analysis of Algorithms:** Analysing control structures: sequencing, "For" loops, Recursive calls, "While" and "repeat" loops, Amortized analysis	04
Unit-III	**Solving Recurrences:** Intelligent guesswork, Homogeneous recurrences, Inhomogeneous Recurrences, Change of variable, Range transformations, Master Theorem, Recurrence Tree.	04
Unit-IV	**Data Structures:** Heaps, Binomial heaps, Disjoint set structures. **Greedy Algorithms:** Graphs: Minimum spanning trees-Kruskal's algorithm, Prim's algorithm, Graphs: Shortest path algorithms.	07
Unit-V	**Divide-and-Conquer:** Multiplying large integers, Binary search, sorting: sorting by merging, quick sort, finding the median, Matrix multiplication, Exponentiation, approaches using recursion, Memory functions. **Dynamic Programming:** Principles of optimality, Various applications using dynamic programming.	08
Unit-VI	**Branch and Bound, Backtracking:** Design of some classical problems using branch and bound and Backtracking approaches. **Randomized and Approximation Algorithms:** Design of some classical problems using Randomized and Approximation Algorithms.	05

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

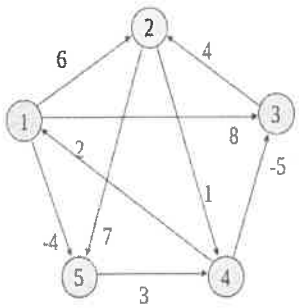
Suggested Readings/ References:

1. Charles E. Leiserson, Thomas H. Cormen, Ronald L. Rivest, Clifford Stein - Introduction to Algorithms, PHI
2. Ellis Horowitz, SartajSahni, Sanguthevar Rajasekharan, Fundamentals of Computer Algorithms,Galgotia.
3. Jean-Paul Tremblay and Paul G. Sorenson, An Introduction to Data Structures with Applications, Tata McGraw Hill
4. Karumanchi, Narasimha, Data Structures and Algorithms Made Easy, CareerMonk Publications.

Suggested List of Experiments:

Sr. No.	Title	Hours
1	To implement iterative and full recursive version of following sorting algorithms: Selection Sort, Insertion Sort and Bubble Sort	02
2	Sort a given set of elements using the quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the 1st to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.	02
3	Sort a given set of elements using Merge sort method and determine the time taken to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.	02
4	Implement Bionomial Heap and all its operations	04
5	 Find Minimum Cost Spanning tree of a given undirected graph using Prim's algorithm and Kruskal's algorithm, and compare the results.	04

	6	Implement all pair Shortest Path problem using Floyd's Algorithm	04



	1	2	3	4	5
1	0	6	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	∞	-5	0	∞
5	∞	∞	∞	3	0

	7	To Implement Knapsack problem based Greedy Algorithm, Dynamic Programming and Branch and Bound Method	04
	8	Matrix chain multiplication problem: Determine the optimal parenthesising of a product of n matrices Matrix A: 10 × 30 Matrix B: 30 × 5 Matrix C: 5 × 60	02
	9	To implement Traveling Salesperson Problem based on Dynamic Programming	02
	10	Implement 8-Queen Problem using Back Tracking	04
Suggested Case List:	-NA-		