GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 3rd Semester
Subject Name: Design and Analysis of Algorithms
Subject Code: 639401

With effective from academic year 2020-21

Prerequisites: Data Structure, C Programming Language.

1. Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE (E)	PA (M)	ESE (V)	PA (I)	
3	-	4	5	70	30	30	20	150

2. Course Outcomes:

Course Outcome Component	Course Outcome (Learner will be able to)
CO1	Analyze algorithms by working out for time and space complexity of algorithms, design algorithm and growth of functions.
CO2	Identify algorithm design methodology to solve problems. Analyze divide-and-conquer, probabilistic analysis and randomized algorithms.
CO3	Able to understand dynamic programming, its elements and analyze existing algorithm of dynamic programming.
CO4	To understand concept of greedy algorithms, amortized analysis and time complexity.
CO5	Apply knowledge of minimum spanning trees and design algorithm for shortest paths

3. Course Duration: The course duration is of 40 sessions of 60 minutes each.

4. Course Contents:

Module No:	Contents	No. of Sessions	70 Marks (External Exam)
Unit I	**Basic Concepts of Analysis and Design of Algorithms in Computing and Growth of Functions:** Algorithms, Algorithms as a technology, Analyzing algorithms, Designing algorithms, Asymptotic notation, Standard notations and common functions.	4	10
Unit II	**Algorithms Using Divide-and-Conquer, Probabilistic Analysis and Randomized Algorithms:** The maximum-subarray problem, Strassen's algorithm for matrix multiplication, The substitution method for solving recurrences, The recursion-tree method for solving recurrences, The master method for solving recurrences, The hiring problem, Indicator random variables, Randomized algorithms.	10	15
Unit III	**Dynamic Programming:** Rod cutting, Matrix-chain	6	12

	multiplication, Elements of dynamic programming, Longest common subsequence, Optimal binary search trees.		
Unit IV	**Greedy Algorithms and Amortized Analysis** An activity-selection problem, Elements of the greedy strategy, Huffman codes. Aggregate analysis, The accounting method, The potential method and Dynamic tables.	10	15
Unit V	**Minimum Spanning Trees and Single-Source Shortest Paths*:*** Growing a minimum spanning tree, The algorithms of Kruskal and Prim, The Bellman-Ford algorithm, Single-source shortest paths in directed acyclic graphs, Dijkstra's algorithm, Difference constraints and shortest paths, Proofs of shortest-paths properties.	10	18

5. Pedagogy:

- ICT enabled Classroom teaching
- Case study
- Practical / live assignment
- Interactive class room discussions

6. Evaluation:

Students shall be evaluated on the following components:

A	**Internal Evaluation**	**(Total - 20 Marks)**
	• Continuous Evaluation Component	10 marks
	• Class Presence & Participation	10 marks
B	**Mid-Semester examination**	**(30 Marks)**
C	**End –Semester Examination(Theory)**	**(70 Marks)**
D	**End –Semester Examination(Practical/Viva)**	**(30 Marks)**

7. Text Book:

No.	Author	Name of the Book	Publisher
1	Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein	Introduction to Algorithms, Third Edition	MIT Press (2009).

8. Reference Books:

No.	Author	Name of the Book	Publisher
1	Parag H Dave, Himanshu B Dave,	Design and Analysis of Algorithms	Pearson (2014)
2	Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein	Introduction to Algorithms, 2nd Edition	PHI
3	Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran	Fundamentals of Computer Algorithms	Universities Press (2008)
4	Anany Levitin	Introduction to Design and Analysis of Algorithms	Pearson (2014)

5	S. Baase	Computer Algorithms: Introduction to Design and Analysis	Pearson (2002)
6	Aho, Hopcroft, and Ullman	The Design and Analysis of Computer Algorithms	Addison Wesley
7	Kleinberg	Algorithm Design	Pearson (2013)

Appendix-A

Sample Practical List

List of Mandatory Lab Exercises (Write Algorithms and Implement in C/C++ Language) for the following problems, students are expected to write one or more (as the case may be) algorithms along with the complexity of these algorithms, and implement them in C/C++ Language.

1. Find square root of a number. Can we use Divide & Conquer approach for this problem?
2. Determine smallest divisor of an integer.
3. For a given value of n, generate prime numbers <= n (more than one algorithms are possible)
4. Find Xn. Iterative and recursive algorithms are possible.
5. Determine product of 2 integers (a * b) as repeated sums. Iterative and recursive algorithms are possible.
6. Find Factorial of n. Iterative and recursive algorithms are possible.
7. Generate Fibonacci series up to n terms Iterative and recursive algorithms are possible.
8. Determine product of 2 large integers using multiplication of their digits. For simplicity, assume both numbers to have same number of digits. This assumption can be relaxed subsequently.
9. Program for finding maximum and minimum number using Divide and conquer.
10. Implement Recursive Binary search and Linear search and determine the time taken to search an element.
11. Breadth First Search (BFS) in a binary tree.
12. Depth First Search (DFS) in a binary tree.
13. Binary Search of an ordered array. Iterative and Recursive algorithms are possible.
14. Sort a given sequence of numbers using (a) Bubble Sort, and (b) Merge Sort
15. Generate permutations of given n numbers. Iterative and recursive algorithms are possible.
16. Find the closest pair out of given n points in 2-dimensional space.
17. Unique partitions of a positive integer.
18. Matrix multiplication using Dynamic Programming algorithm.
19. Solution of Rod-cutting problem using Dynamic Programming algorithm.
20. Generate pseudo-random numbers.
21. Implement Strassen’s algorithm for matrix multiplication
22. Prim's algorithm to find minimum cost tree (shortest path in a tree).
23. Kruskal's algorithm to find minimum cost tree (shortest path in a tree).
24. Implement Bellman-Ford Single Source Shortest Path Algorithm
25. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm