# GUJARAT TECHNOLOGICAL UNIVERSITY

**Bachelor of Engineering**

**Subject Code: 3134201**

Data Structures and Algorithms

3rd Semester

**Type of course:** Core

**Prerequisite:** Fundamentals of programming and problem solving, Engineering mathematics

**Rationale:**

- To understand the basic concepts of data structures and algorithms.
- To differentiate linear and non-linear data structures and the operations upon them.
- Ability to perform sorting and searching in each set of data items.
- To comprehend the necessity of time complexity in algorithms.

**Teaching and Examination Scheme:**

| Teaching Scheme | | | Credits | Examination Marks | | | | Total Marks |
|---|---|---|---|---|---|---|---|---|
| | | | | Theory Marks | | Practical Marks | | |
| L | T | P | C | ESE (E) | PA | ESE (V) | PA (I) | |
| 3 | 0 | 4 | 5 | 70 | 30 | 30 | 20 | 150 |

**Contents:**

| Sr. No. | Content | Total Hrs |
|---|---|---|
| 1 | **Introduction to Algorithms and Analysis:** Overview and importance of algorithms and data structures. Fundamentals of algorithm analysis, Space and time complexity of an algorithm, Types of asymptotic notations and orders of growth, Algorithm efficiency – best case, worst case, average case, Analysis of non-recursive and recursive algorithms, Asymptotic analysis for recurrence relation – Recursive Tree Method. | 06 |
| 2 | **Linear Data Structures:** Array- 1D and 2D array, Stack - Applications of stack: Expression Evaluation - Conversion of Infix to postfix and prefix expression, Tower of Hanoi. Queue - Types of Queue: Circular Queue, Double Ended Queue (deQueue), Applications – Priority Queue using Arrays, List - Singly linked lists – Doubly linked lists - Circular linked lists. | 08 |
| 3 | **Sorting and Searching Techniques, Hashing:** Searching - Linear Search and binary search (divide and conquer technique), Applications - Sorting – Insertion sort - Selection sort – Bubble sort – Quick sort – Merge sort, Analysis of sorting algorithms. Hash functions, open hashing-separate chaining, closed hashing - linear probing, quadratic probing, double hashing | 07 |
| 4 | **Divide and Conquer Algorithm:** Introduction, Recurrence, and different methods to solve recurrence, multiplying large Integers Problem. | 04 |
| 5 | **Non-linear Data Structures – Trees:** Tree - Terminology, Binary Tree – Terminology and Properties, Tree Traversals, Expression Trees – Binary Search Trees – operations in BST – insertion, deletion, finding min and max, Finding the kth minimum element in a BST, AVL tree. | 07 |
| 6 | **Non-linear Data Structures – Graphs:** Graph – basic definition and Terminology – Representation of Graph – Graph Traversal: Breadth First Search (BFS), Depth First Search (DFS) | 04 |
| 7 | **Greedy Algorithm:** General Characteristics of greedy algorithms, Problem solving using Greedy Algorithm - Activity selection problem, Elements of Greedy Strategy, | 04 |

| | | |
|---|---|---|
| | Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Graphs: Shortest paths, The Knapsack Problem, Job Scheduling Problem | |
| 8 | **Dynamic Programming:** Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming – Calculating the Binomial Coefficient, Making Change Problem, Assembly Line-Scheduling, Knapsack problem, Matrix chain multiplication, Longest Common Subsequence. | 05 |
| 9 | **Backtracking and Branch and Bound:** Introduction, The Eight queens' problem, Knapsack problem, Travelling Salesman problem, Minimax principle | 03 |

**Suggested Specification table with Marks (Theory):**

| **Distribution of Theory Marks** | | | | | |
|---|---|---|---|---|---|
| R Level | U Level | A Level | N Level | E Level | C Level |
| **10** | **25** | **25** | **10** | **--** | **--** |

**Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create and above Levels (Revised Bloom's Taxonomy)**

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary from above table.

**Reference Books:**

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, PHI.
2. Fundamentals of Algorithms – E. Horowitz et al.
3. Horowitz, Sahni, and S. Anderson-Freed, Fundamentals of Data Structures in C, Universities Press, Second Edition,2008.
4. Data Structures using C, Reema Thareja, Oxford University Press, Second Edition.
5. Introduction to Design and Analysis of Algorithms, Anany Levitin, Pearson.

**Course Outcomes:** Students will be able to

| **Sr. No.** | **CO Statement** | **Marks % Weightage** |
|---|---|---|
| 1 | Understanding the fundamental analysis and asymptotic performance of algorithms. | 30 |
| 2 | Derive and solve recurrences describing the performance of algorithms. | 15 |
| 3 | Application of appropriate data structures and algorithms to find solutions to practical problems. | 35 |
| 4 | Find optimal solution by applying various methods. | 20 |

**List of Experiments:**

**(Pl. Note: List of Experiments and Tutorials should be as per theory covered in the class, below mentioned practical are just for the reference purpose)**

1. Implementation and time analysis of Stack and its applications (infix, postfix, prefix).
2. Implementation and time analysis of queue and its applications.
3. Implementation and time analysis of singly linked list.

4. Implementation and time analysis of doubly linked list.
5. Implementation and time analysis of circular linked list.
6. Implementation of binary tree and its traversal (preorder, inorder, postorder).
7. Implementation and Time analysis of sorting algorithms – Bubble sort, Selection sort, Insertion sort, Merge sort and Quicksort.
8. Implementation and Time analysis of linear and binary search algorithm.
9. Implementation and Time analysis of factorial program using iterative and recursive method
10. Implementation of a knapsack problem using dynamic programming.
11. Implementation of chain matrix multiplication using dynamic programming.
12. Implementation of a knapsack problem using greedy algorithm
13. Implementation of Graph and Searching (DFS and BFS).
14. Implement prim's algorithm
15. Implement 3ruskal's algorithm.
16. Implement LCS problem.