Course Code: BTHPCCCST301, BTHPCCCSI301
Course Title: Data Structure & Algorithms
Credit: 3

Module I:
Introduction:
Why we need data structure?
Concepts of data structures: a) Data and data structure b) Abstract Data Type and Data Type.
Applications
Algorithms and programs, basic idea of pseudo-code.
Algorithm efficiency and analysis, time and space complexity analysis of algorithms – order notations.

Module 2: Linear Data Structures
Array:
Different representations – row major, column major.
Sparse matrix - its implementation and usage. Array representation of polynomials.
Linked List:
Singly linked list, circular linked list, doubly linked list, linked list representation of polynomials and applications.
Stack and Queue:
Stack and its implementations (using array, using linked list), applications (Infix to postfix conversion, Evaluation of Postfix expression etc.).
Queue, circular queue, dequeue. Implementation of queue- both linear and circular (using array, using linked list), applications.
Recursion:
Principles of recursion – use of stack, differences between recursion and iteration, tail recursion.
Applications - Tower of Hanoi.

Module 3: Nonlinear Data structures:
Trees:
Basic terminologies, tree representation (using array, using linked list).
Binary trees - binary tree traversal (pre-, in-, post- order), recursive and non-recursive traversal algorithms of binary tree, threaded binary tree (left, right, full), and expression tree.
Binary search tree- operations (creation, insertion, deletion, searching).
Height balanced binary tree – AVL tree (insertion, deletion with examples only).
B- Trees – operations (insertion, deletion with examples only).
B+ Trees – operations (insertion, deletion with examples only).
Graphs:
Graph definitions and concepts (directed/undirected graph, weighted/un-weighted edges, sub-graph, degree, cut vertex/ articulation point, pendant node, clique, complete graph, connected components – strongly connected component, weakly connected component, path, shortest path, and isomorphism). Graph representations/storage implementations – adjacency matrix, adjacency list, adjacency multi-list. Graph traversal and connectivity – Depth-first search (DFS), Breadth-first search (BFS) – concepts of edges used in DFS and BFS (tree-edge, back-edge, cross-edge, forward-edge), applications.
Minimal spanning tree – Prim's algorithm, Kruskal's algorithm (basic idea of greedy methods).

Module 4:

Searching and Sorting:

Sorting Algorithms: Bubble sort, insertion sort, shell sort, selection sort, merge sort, quick sort, heap sort (concept of max heap, application – priority queue), radix sort.

Time and space complexity derivations.

Searching: Sequential search, binary search, interpolation search.

Time and space complexity derivations.

Hashing: Hashing functions, collision resolution techniques.