ADVANCE DATA STRUCTURES

I Semester: CSE

Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
BCSB02	Core	3	-	-	3	30	70	100
Contact Classes:45	Tutorial Classes: Nil	Practical Classes: Nil				Total Classes: 45		

COURSE OBJECTIVES:

The course should enable the students to:

I Understand the data structures and techniques of algorithm analysis.

II Solve problems using different data structures and compare their performance and tradeoffs.

III Illustrate the implementation of linked data structures such as linked lists and binary trees.

IV Understand graph algorithms such as shortest path and minimum spanning tree.

V Learn advanced data structures such as balanced search trees, hash tables, priority queues

COURSE OUTCOMES (COs):

CO 1: Implementation of hash tables, including collision avoidance and resolution schemes.
CO 2: Analyze how to balance a binary search tree using rotation methods and color changing methods.
CO 3: Solve problems using graph algorithms, including single-source and all-pairs shortest paths,and minimum spanning tree algorithms.
CO 4: Relates all binary heap trees to form a large binomial queue for large data structures creation.
CO 5: Reconstructs such applications that take the advantage of a trie's ability to quickly search for, insert, and delete entries into the dictionary.

COURSE LEARNING OUTCOMES (CLOs):

1. Analyze time and space complexity of an algorithm for their performance analysis
2. Understand arrays, single and doubly linked lists in linear data structure and tress , graphs in non-linear data structure
3. Master a variety of advanced abstract data type (ADT) and their implementations
4. Understand dynamic data structures and relevant standard algorithms
5. Design and analyze and Concepts of heap, priority queue
6. Analyze probing methods like linear probing and quadratic probing
7. Understand and implement hash table and linear list representation
8. Understand the properties of binary tress and implement recursive and non-recursive traversals
9. Understand graphs terminology, representations and traversals in Graphs
10. Implement Depth First Search and Breath First Searching methods of non –linear data structures
11. Analyze Dijkstra's algorithm for single source shortest path problem for minimum cost spanning trees
12. Implement binary search ADT for finding parent node, smallest and largest values in binary search
13. Understand and implement operations and applications of red-Black and splay Trees
14. Implement Huffman Coding and decoding for text compression.

UNIT- I	OVERVIEW OF DATA STRUCTURES	Classes: 09

Algorithm analysis: Algorithms; Performance analysis: Time complexity and space complexity, asymptotic notation: Big Oh, omega and theta notations, complexity analysis examples; Data structures: Linear and non linear data structures, ADT concept, linear list ADT, stack and queue ADTs, array and linked list representations; Circular queue: Insertion and deletion, de queue ADT, priority queue ADT, implementation

using heaps, insertion into a max heap, deletion from a max heap, singly linked lists, doubly linked lists, circular linked list.

UNIT - II	DICTIONARIES, HASH TABLES	Classes: 09

Dictionaries: Linear list representation, operations insertion, deletion and searching, hash table representation, hash functions, collision resolution, separate chaining, open addressing, linear probing, quadratic probing, double hashing, rehashing, extendible hashing

UNIT - III	TREES AND GRAPHS	Classes: 09

Trees: Ordinary and binary trees terminology, properties of binary trees, binary tree ADT, representations, recursive and non recursive traversals, threaded binary trees.

Graphs: Graphs terminology, graph ADT, representations, graph traversals; Search methods: DFS and BFS; Applications of Graphs: Minimum cost spanning tree using Kruskal‟s algorithm, Dijkstra‟s algorithm for single source shortest path problem.

UNIT - IV	SEARCH TREES I	Classes: 09

Binary search tree: Binary search tree ADT, insertion, deletion and searching operations, finding the parent of a given node, attaining a reference to a node, finding the smallest and largest values in the binary search tree.

Balanced search trees: AVL trees, definition, height of an AVL tree; Operations : Insertion, deletion and searching.

UNIT - V	SEARCH TREES II	Classes: 09

Red-Black and Splay Trees; B trees: Definition, operations and applications; R trees: Nearest neighbor query, join and range queries; Comparison of search trees; Text compression: Huffman coding and decoding; Pattern matching: KMP algorithm.

Text Books:

1. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, Universities Press Private Limited, India, 2nd Edition, 2008.
2. G.A. V.Pai, “Data Structures and Algorithms”, Tata McGraw Hill, New Delhi, 1st Edition, 2008.
3. M. A. Weiss, Addison Wesley, “Data Structures and Algorithm Analysis in Java”, Pearson Education,2nd Edition, 2005.

Reference Books:

1. D. Samanta, “Classic Data Structures”, Prentice Hall of India Private Limited, 2nd Edition, 2003.
2. Aho, Hop craft, Ullman, “Design and Analysis of Computer Algorithms”, Pearson Education India,1st Edition, 1998.
3. Goodman, Hedetniemi, “Introduction tothe Design and Analysis of Algorithms”, Tata McGraw Hill, New Delhi, India, 1st Edition, 2002.
4. Adam Drozdek, “Data Structures and Algorithms in C++”, Thomson Course Technology, 3rd Edition, 2005.
5. M. T. Goodrich, R. Tomassia, “Data structures and Algorithms in Java”, Wiley India, 3rd Edition, 2011.

Web References:

1. http://www.tutorialspoint.com/data_structures_algorithms/data_structures_basics.htm
2. http://www.geeksforgeeks.org/b-tree-set-1-introduction-2/
3. http://www.nptelvideos.in/2012/11/data-structures-and-algorithms.html

E-Text Books:

1. https://comsciers.files.wordpress.com/2015/12/horowitz- -of-computer-algorithms-2nd-edition.pdf