

**II Year - II Semester**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>4</b>	<b>0</b>	<b>0</b>	<b>3</b>

## **ADVANCED DATA STRUCTURES**

### **OBJECTIVES:**

- Describe and implement a variety of advanced data structures (hash tables, priority queues, balanced search trees, graphs).
- Analyze the space and time complexity of the algorithms studied in the course.
- Identify different solutions for a given problem; analyze advantages and disadvantages to different solutions.
- Demonstrate an understanding of external memory and external search and sorting algorithms.
- Demonstrate an understanding of simple Entity-Relationship models for databases.

### **UNIT-I: SORTING**

External Sorting, Introduction, K-way Merging - Buffer Handling for parallel Operation- Run Generation- Optimal Merging of Runs.

### **UNIT-II: HASHING**

Introduction-Static Hashing- Hash Table- Hash Functions- Secure Hash Function- Overflow Handling- Theoretical Evaluation of Overflow Techniques, Dynamic Hashing- Motivation for Dynamic Hashing -Dynamic Hashing Using Directories- Directory less Dynamic, Hashing,

### **UNIT-III:PRIORITY QUEUES (HEAPS)**

Model, Simple Implementation, Binary Heap-Structure Property-Heap-Order Property-Basic Heap Operations- Other Heap Operation, Applications of Priority Queues- The Selection Problem Event Simulation Problem, Binomial Queues- Binomial Queue Structure – Binomial Queue Operation- Implementation of Binomial Queues

### **UNIT-IV: EFFICIENT BINARY SEARCH TREES**

Optimal Binary Search Trees, AVL Trees, Red-Black Trees, Definition- Representation of a Red- Black Tree- Searching a Red-Black Tree- Inserting into a Red Black Tree- Deletion from a Red-Black Tree- Joining Red-Black Trees, Splitting a Red-Black tree.

### **UNIT-V: MULTIWAY SEARCH TREES**

M-Way Search Trees, Definition and Properties- Searching an M-Way Search Tree, B-Trees, Definition and Properties- Number of Elements in a B-tree- Insertion into B-Tree- Deletion from a B-Tree- B+-Tree Definition- Searching a B+-Tree- Insertion into B+-tree- Deletion from a B+-Tree.

## **UNIT-VI: DIGITAL SEARCH STRUCTURES**

Digital Search Trees, Definition- Search, Insert and Delete- Binary tries and Patricia, Binary Tries, Compressed Binary Tries- Patricia, Multiway Tries- Definitions- Searching a Trie- Sampling Strategies- Insertion into a Trie- Deletion from a Trie- Keys with Different Length- Height of a Trie- Space Required and Alternative Node Structure- Prefix Search and Applications- Compressed Tries- Compressed Tries With Skip Fields- Compressed Tries With Labeled Edges- Space Required by a Compressed Tries, Tries and Internet Packet Forwarding , - IP Routing- 1-Bit Tries- Fixed-Stride Tries-Variable-Stride Tries.

### **OUTCOMES:**

- Be able to understand and apply amortised analysis on data structures, including binary search trees, mergable heaps, and disjoint sets.
- Understand the implementation and complexity analysis of fundamental algorithms such as RSA, primality testing, max flow, discrete Fourier transform.
- Have an idea of applications of algorithms in a variety of areas, including linear programming and duality, string matching, game-theory

### **TEXT BOOKS:**

1. Data Structures, a Pseudocode Approach, Richard F Gilberg, Behrouz A Forouzan, Cengage.
2. Fundamentals of DATA STRUCTURES in C: 2<sup>nd</sup> ed, , Horowitz , Sahani, Anderson-freed, Universities Press
3. Data structures and Algorithm Analysis in C, 2<sup>nd</sup> edition, Mark Allen Weiss, Pearson

### **REFERENCE BOOKS:**

1. Web : <http://lcm.csa.iisc.ernet.in/dsa/dsa.html>
2. [http://utubersity.com/?page\\_id=878](http://utubersity.com/?page_id=878)
3. <http://freevideolectures.com/Course/2519/C-Programming-and-Data-Structures>
4. <http://freevideolectures.com/Course/2279/Data-Structures-And-Algorithms>
5. File Structures :An Object oriented approach with C++, 3<sup>rd</sup> ed, Michel J Folk, Greg Riccardi, Bill Zoellick
6. C and Data Structures: A Snap Shot oriented Treatise with Live examples from Science and Engineering, NB Venkateswarlu & EV Prasad, S Chand, 2010.