Course Structure for Computer Science Engineering

Semester III

Sr. No.	Subject Name	L	T	P	Credits	Total
1	Data Structures	3	0	4	5	150
2	Computer Organization &Architecture	3	0	0	3	100
3	Introduction to Processors and Chips	3	0	2	4	125
4	Discrete Mathematics	3	1	0	4	100
5	Python Programming Lab	0	0	4	2	50
6	Mini Project –I (Python)	0	0	2	1	100
7	Economics and Finance for Engineers	3	0	0	3	100
8	Environmental Studies *	1	0	0	1*	50
Total		**15**	**1**	**12**	**22**	**775**

Semester IV

Sr. No.	Subject Name	L	T	P	Credits	Total
1	Applied Mathematics	3	1	0	4	100
2	Advanced Data Structures	3	0	4	5	150
3	Computer Networks	3	0	2	4	125
4	Software Engineering and Project Management	3	0	0	3	100
5	Java Programming Lab	0	0	4	2	50
6	Mini Project-II(Java)	0	0	2	1	100
7	Theory of Computation	3	1	0	4	100
Total		**15**	**2**	**12**	**23**	**725**

MIT- ADT University, Pune

MIT School of Engineering

B.Tech (SY-CSE) SEMESTER IV

Course Code	Course Title	Category
18BTCS402	Advanced Data Structures & Algorithms	**Core**

Contact Hours per Week			CA	FE	Credits
L	T	D/P			
3	1	4	**40**	**60**	**4**

Prerequisite: Discrete Mathematics, Data Structures

Course Objectives:

1. To suggest an appropriate non-linear data structures and algorithm for graphical solutions of the problems.
2. To understand the various searching and indexing techniques.
3. To suggest appropriate data structure and algorithm for graphical solutions of the problems.
4. To learn various file operations relevant to different types of files.
5. To develop the logical skills and use of appropriate data structures in different domains

COURSE CONTENT

I. TREES 9

Tree- basic terminology, General tree and its representation, representation using sequential and linked organization, Binary tree- properties, converting tree to binary tree, binary tree traversals in-order, pre-order, post order, level wise -depth first and breadth first, Operations on binary tree. Binary Search Tree (BST), BST operations, Threaded binary tree- concepts, threading, insertion and deletion of nodes in in-order threaded binary tree, in order traversal of in-order threaded binary tree.

Case Study- Use of binary tree in expression tree-evaluation and Huffman's coding

II. GRAPHS 9

Basic Concepts, Storage representation, Adjacency matrix, adjacency list, adjacency multi list, inverse adjacency list. Traversals-depth first and breadth first, Introduction to

Greedy Strategy, Minimum spanning Tree, Greedy algorithms for computing minimum spanning tree- Prim’s and Kruskal’s Algorithms, Dijktra's Single Source shortest path, Topological ordering.

Case study- Data structure used in Webgraph and Google map.

III.HASHING 9

Hash Table- Concepts-hash table, hash function, bucket, collision, probe, synonym, overflow, open hashing, closed hashing, perfect hash function, load density, full table, load factor, rehashing, issues in hashing, hash functions- properties of good hash function, division, multiplication, extraction, mid-square, folding and universal, Collision resolution strategies- open addressing and chaining, Hash table overflow- open addressing and chaining, extended hashing, Dictionary- Dictionary as ADT, ordered dictionaries, Skip List- representation, searching and operations- insertion, removal.

IV. SEARCH TREES, INDEXING AND MULTIWAY TREES 9

Symbol Table-Representation of Symbol Tables- Static tree table and Dynamic tree table, Introduction to Dynamic Programming, Weight balanced tree, introduction to Optimal Binary Search Tree (OBST), Height Balanced Tree AVL tree. Indexing and multi way Trees- Indexing, indexing techniques, Types of search tree- multiway search tree, B-Tree, B+ Tree, Trie Tree, Splay Tree, Red-Black Tree, K dimensional tree, ~~AA tree~~, Heap-Basic concepts, realization of heap and operations, Heap as a priority queue, heap sort

Case study: OBST as an example of Dynamic Programming,

V. FILE ORGANIZATION 9

Sequential file organization- concept and primitive operations, Direct Access File Concepts and Primitive operations, Indexed sequential file organization-concept, types of indices, structure of index sequential file, Linked Organization- multi-list files, coral rings, inverted files and cellular partitions. External Sort- Consequential processing and merging two lists, multi day merging- a k-way merge algorithm.

Course Outcomes:
Students will be able to:

1. Build solutions using various non-linear data structures.
2. Analyze and implement various searching and indexing techniques.
3. Integrate file creation and primitive operations for various file types.
4. Analyze the algorithmic solutions for resource requirements and optimization

---

TEXT BOOKS

1. Horowitz, Sahani, Dinesh Mehata, ―Fundamentals of Data Structures in C++‖, Galgotia Publisher, ISBN: 8175152788, 9788175152786.
2. M Folk, B Zoellick, G. Riccardi, ―File Structures‖, Pearson Education, ISBN:81- 7758-37-5
3. Peter Brass, ―Advanced Data Structures‖, Cambridge University Press, ISBN: 978- 1-

107-43982-5

4. Varsha Patil -- Data Structures Using C++ , Oxford University Press, 978-0-198- 06623-1

REFERENCES

1. A. Aho, J. Hopcroft, J. Ulman, —Data Structures and Algorithms‖, Pearson Education, 1998,ISBN-0-201-43578-0.
2. Michael J Folk, —File Structures an Object Oriented Approach with C++‖, Pearson Education, ISBN: 81-7758-373-5.
3. SartajSahani, —Data Structures, Algorithms and Applications in C++‖, Second Edition, University Press, ISBN:81-7371522 X.
4. G A V Pai, —Data Structures and Algorithms‖, The McGraw-Hill Companies, ISBN -9780070667266.
5. Goodrich, Tamassia, Goldwasser, —Data Structures and Algorithms in Java‖, Wiley Publication, ISBN: 9788126551903.

Course Code	Course Title				Category
18BTCS411	**Advance Data Structures Laboratory**				**Core**
Contact Hours per Week			**A**	**FE**	**Credits**
L	**T**	**D/P**			
0	**0**	**4**	**40**	**60**	**2**
Prerequisite: C/C++/ Basic python programming					
Course Objectives: 1. To Learn non-linear data structures for developing applications. 2. To understand the hashing techniques for performing searching in real-time applications 3. To learn data storing techniques for performing data management.					

B.Tech. CSE – Second year

COURSE CONTENT

Part A – Advanced Data Structures and Algorithms

List of Experiments

1.	Accept prefix expressions, and construct a binary tree and perform recursive and non-recursive traversals.
2.	A Dictionary stores keywords & its meanings. Provide facility for adding new keywords, deleting keywords, updating values of any entry. Provide a facility to display whole data sorted in ascending/ Descending order. Also, find how many maximum comparisons may require for finding any keyword. Use Binary Search Tree for implementation.
3.	Create a Binary Search tree and find its mirror image. Print original & new tree level wise. Find height & print leaf nodes.
4.	Create an inorder threaded binary search tree and perform the traversals.
5.	Represent a given graph using an adjacency list and perform DFS or BFS.
6.	Represent a given graph using an adjacency list or array and find the shortest path using Dijkstra’s algorithm.

7.	Represent a given graph using an adjacency list or array and generate a minimum spanning tree using Kruskal's or Prim’s algorithm.
8.	Create a hash table and handle the collisions using linear probing with or without replacement
9.	Implementation of simple index file
10.	Implementation of direct access file - Insertion and deletion of a record from a direct access file using chaining with or without replacement
11.	Company maintains employee information such as employee ID, name, designation and salary. Allow users to add, delete information about employees. Display information of a particular employee. If an employee does not exist, an appropriate message is displayed. If it is, then the system displays the employee details. Use a sequential file to maintain the data.
12.	A Dictionary stores keywords & its meanings. Provide facility for adding new keywords, deleting keywords, updating values of any entry. Provide a facility to display whole data sorted in ascending/ Descending order. Also, find how many maximum comparisons may require for finding any keyword. Use Height balanced tree and find the complexity for finding a keyword
13.	Consider the telephone book database of N clients. Make use of a hash table implementation to quickly look up client‘s telephone number
14.	Implement Heap sort.
15.	Given sequence k = k1 <k2 < … <kn of n sorted keys, with a search probability pi for each key ki . Build the Binary search tree that has the least search cost given the access probability for each key? ----wants to remove

TEXT BOOKS

1. Horowitz, Sahani, Dinesh Mehata, —Fundamentals of Data Structures in C++‖, Galgotia Publisher, ISBN: 8175152788, 9788175152786.

2. M Folk, B Zoellick, G. Riccardi, —File Structures‖, Pearson Education, ISBN:81-7758-37-5

3. Peter Brass, —Advanced Data Structures‖, Cambridge University Press, ISBN: 978-1-107-43982-5

REFERENCES

1. A. Aho, J. Hopcroft, J. Ulman, ―Data Structures and Algorithms‖, Pearson Education, 1998,ISBN-0-201-43578-0.

2. Michael J Folk, ―File Structures an Object Oriented Approach with C++‖, Pearson Education, ISBN: 81-7758-373-5.

3. Sartaj Sahani, ―Data Structures, Algorithms and Applications in C++‖, Second Edition, University Press, ISBN:81-7371522 X.

4. G A V Pai, ―Data Structures and Algorithms‖, The McGraw-Hill Companies, ISBN - 9780070667266.

5. Goodrich, Tamassia, Goldwasser, ―Data Structures and Algorithms in Java‖, Wiley Publication, ISBN: 9788126551903.

Course Outcomes:

Students will be able to –

1. Create applications using non-linear data structures
2. Perform hashing operations for searching
3. Build solutions for data storage problems.