Subject Title: Advanced Data Structures					
Course Code: 19CST221 /19ITT221	**Year and Semester:** II Year II semester	L	T	P	C
Prerequisites: Prior knowledge of programming language(s) and basic Data Structures and Algorithms		2	2	0	3

Course Objectives:

1. To impart the knowledge on various hashing techniques.
2. To help the students to learn Priority Queues and its applications.
3. To demonstrate the students about the operations of Efficient Search Trees.
4. To make the student to understand various shortest path algorithms in graphs.
5. To make the students to learn the use of Digital Search Structures and pattern matching algorithms.

Course Outcomes:

At the end of the course student will be able to:

CO1: Able to understand the implementation of symbol table using hashing techniques. **(Understand)**

CO2: Able to understand the trade-offs of implementation of priority queues. **(Understand)**

CO3: Able to understand and implement efficient search trees. **(Understand)**

CO4: Able to appreciate the importance and significance of graph algorithms in building and solving real world applications. **(Analyze)**

CO5: Able to understand and implement algorithms for pattern matching in a text. **(Understand)**

UNIT - I **12 Hours**

Hashing: Static Hashing, Hash Functions, , Secure Hash Functions, Collision Resolution Techniques-, Dynamic Hashing.

UNIT - II **12 Hours**

Priority Queues (Heaps): Binary Heaps, Applications of Heap, Binomial Heaps.

UNIT - III **14 Hours**

Efficient Search Trees: Self-balancing Binary Search Trees, AVL Tree, Red-Black Tree, B-tree, B+ Tree.

UNIT - IV **10 Hours**

Graph Algorithms : Elementary Graph Operations, Graph Algorithms: Minimal Cost Spanning Tree, Shortest Path.

UNIT - V **12 Hours**

Tries: Digital Search Trees(Tries), Operations, Different types of Tries **Pattern Matching Algorithms**.

Text Books:

1. Advanced Data Structures, Reema Thareja, S. Rama Sree, Oxford University Press, 2017.
2. Fundamentals of Data Structures in C, Horowitz, Sahni, Anderson-Freed, Second Edition, 2008.

Reference Books:

1. Advanced Data Structures, Peter Brass, Cambridge University Press, 2008.
2. Data Structures and Algorithms, A. V. Aho, J. E. Hopcroft, and J. D. Ullman, Pearson, 2002.
3. Data Structures and Algorithm Analysis in C, Mark Allen Weiss, Second Edition, Pearson.

Micro Syllabus of Advanced Data Structures

II B. Tech II Semester

UNIT I :
Hashing: Static Hashing, Hash Functions, Secure Hash Functions, Collision Resolution Techniques-, Dynamic Hashing.

Unit	Module	Micro Content
UNIT I	Hash Functions, Securing Hash Functions	Hash Function
		Division Method
		Mid Square Method
		Folding Method
		Universal Hash Function
	Collision Resolution Techniques	Linear Probing
		Quadratic Probing
		Double Hashing
		Rehashing
		Separate Chaining
	Dynamic Hashing	Introduction
		Dynamic Hashing using Directories
		Directory less Dynamic Hashing
	Additional Topics	Implementation of Hash Table

UNIT - II

Priority Queues: Binary Heaps, Applications of Heap, Binomial Heaps.

Unit	Module	Micro Content
UNIT II	Priority Queue	Introduction, Types of Priority Queues
	Binary Heap	Properties of Heaps, Representation
		Operations: Insert, Delete, ExtractMin, BuildHeap, Reheapify
		Applications of Heap : Heap Sort, Prims Algortihm
	Binomial Heap	Properties of Binomial Tree, Representation
		Operations on Binomial Heap: Insert, Extract Min, Delete, DecreaseKey
		Applications of Binomial Heap
	Additional Topics	Huffman Coding

UNIT - III

Efficient Search Trees: Self-balancing Binary Search Trees, AVL Tree, Red-Black Tree, B-tree, B+ Tree.

Unit	Module	Micro Content
UNIT III	Self Balancing Binary Search Trees (BSTs)	Binary Search Trees recap
		Necessity of Self Balancing BST
	AVL Trees	Properties of AVL Trees, Representation
		Rotations of AVL Trees : LL, RR, LR, and RL rotations.
		Operations: Insert, Delete, search
	Red – Black Tree	Properties of Red – Black Tree, Representation
		Operations : Insertion, Deletion
	B – Tree	Properties of B – Tree, representation
		Operations on B – Tree: Insertion, Deletion, Search
	B^+ Tree	Properties of B^+ Tree, representation
		Operations on B^+ Tree: Insertion, Deletion, Search
	Additional Topic	AVL Sort

UNIT - IV

Graph Algorithms : Elementary Graph Operations, Graph Algorithms: Minimal Cost Spanning Tree, Shortest Path.

Unit	Module	Micro Content
UNIT IV	Elementary Graph Operations	Recap of Graphs

		Connected Components, Bi connected components
	Minimal Cost Spanning Tree Algorithms	Prims Algorithm (recap)
		Kruskal’s Algorithm (recap)
		Sollin’s (Boruvka’s) Algortihm
	Shortest Path Algorithm	Dijkstra’s Algorithm
		Bellman Ford Algorithm
	All Pair Shortest Path Algorithms	Transitive Closure : Warshall’s Algorithm
		Shortest Path: Floyd’s Algorithm
	Additional Topic	Implementation of DAG

UNIT V
Tries: Digital Search Trees(Tries), Operations, Different types of Tries
Pattern Matching Algorithms.

Unit	Module	Micro Content
UNIT V	Tries	Digital Search Trees
		Operations on Trie: Insertion, Deletion, Search
		Binary Tries
		Compressed Binary Trie
		Suffix Trie
		PATRICIA
	Pattern Matching Algorithms	Brute Force Algorithm
		Boyer Moore Algorithm
		Knuth Morris Pratt Algorithm
	Additional Topic	Rabin – Karp Algorithm

Advanced Data Structures Lab					
Course Code: **19CSP221/ 19ITP221**	Year and Semester: II-II	L	T	P	C
Prerequisites: Prior knowledge of programming language(s) and basic Data Structures and Algorithms		0	0	3	1.5

Course Objectives:

1. Ability to apply computational thinking to a diverse set of problems.
2. Ability to adapt to new challenges and computational environments.
3. Proficiency in the design and implementation of algorithms.

Course Outcomes:

At the end of the course student will be able to:

1. CO1: Select the most appropriate data structure and defend the selection.
2. CO2: Appropriately solve a variety of computational problems.
3. CO3: Communicate their results and describe an algorithm.

List of experiments:

Prerequisites : Solve the following problems in Hackerrank

1. Time Conversion
2. Balanced Brackets
3. Queue using 2 Stacks
4. Cycle Detection

UNIT I

1 Implement a dictionary (Division Method)

2 Implement Linear Probing on a dictionary

3 Implement Quadratic Probing on a dictionary

4 Implement Double Hashing.

5 Implement Separate Chaining.

6 **String Pairs / Anagram**

UNIT II

7 Implement Binary Heap Operations.

8 **Minimize the Sum**

9 **Implement Expression Tree.**

UNIT III

10 Implement Operations on Binary Search Tree non recursive.

11 Implement AVL Tree.

UNIT IV

12 Implement Prims Algorithm

13 Implement Krushkal’s Algorithm

14 Implement Sollin’s Algorithm

15 Implement Dijkstra’s Algorithm

16 Implement shortest path between all pair of vertices.

17 **Implement Island Strikes.**

18 **Implement Pawn Moves.**

UNIT V

19 Implement Brute force pattern matching algorithm.

20 Implement Boyer-Moore pattern matching algorithm.

21 Implement Knuth-Morris pattern matching algorithm.

22 **Implement Counting Numeric sub sequences.**