INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad -500 043

COMPUTER SCIENCE AND ENGINEERING

COURSE DESCRIPTOR

Course Title	**ADVANCED DATA STRUCTURES LABORATORY**				
Course Code	BCSB09				
Programme	M.Tech				
Semester	I	CSE			
Course Type	Core				
Regulation	IARE - R18				
Course Structure	**Theory**			**Practical**	
	Lectures	**Tutorials**	**Credits**	**Laboratory**	**Credits**
	-	-	-	3	2
Course Faculty	Ms. S Swarajya Laxmi, Assistant Professor, CSE				

I. COURSE OVERVIEW:

This course helps the students to implement linear and non linear data structures to solve various computing problems. This includes algorithms and analysis of algorithms based on their time complexity, implementation of applications of various data structures like stacks, trees and graphs.

II. COURSE PRE-REQUISITES:

Level	Course Code	Semester	Prerequisites	Credits
-	-	-	Basic Programming Concepts	-

III. MARKS DISTRIBUTION:

Subject	SEE Examination	CIA Examination	Total Marks
Advanced Data Structures Laboratory	70 Marks	30 Marks	100

IV. DELIVERY / INSTRUCTIONAL METHODOLOGIES:

✔	LCD / PPT	✔	Student viva	✔	Mini Project	✗	Videos
✗	Open Ended Experiments						

V. EVALUATION METHODOLOGY:

Continuous internal assessment (CIA):

The course will be evaluated for a total of 100 marks consisting of 30 marks for Continuous Internal Assessment (CIA) and 70 marks for Semester End Examination (SEE). Out of 30 marks allotted for CIA during the semester, with 20 marks for the day to day evaluation and 10 marks for Internal Examination (CIE).

Semester End Examination (SEE):

The SEE is conducted for 70 marks of 3 hours duration. The syllabus for this course contains 12 experiments. The question paper pattern is as follows: Two full questions with 'either' 'or' choice will be drawn from each set. Each contains 4 questions

Continuous internal assessment (CIA):

CIA is conducted for a total of 30 marks (Table 1), with 10 marks for Continuous Internal Examination (CIE),20 marks for Day to Day Evaluation

Table 1: Assessment pattern for CIA

Component	**Theory**		**Total Marks**
Type of Assessment	CIE Exam	Day to Day Evaluation	
CIA Marks	10	20	30

Continuous Internal Examination (CIE):

Two CIE exams shall be conducted at the end of the 16th week of the semester. The CIE exam is conducted for 10 marks of 3 hours duration consisting of two sets.

Preparation	**Performance**	**Calculations and Graph**	**Results and Error Analysis**	**Viva**	**Total**
2	2	2	2	2	10

VI. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes (POs)		**Strength**	**Proficiency assessed by**
PO 1	Analyze a problem, identify and define computing requirements, design and implement appropriate solution	2	Laboratory practices, student viva
PO 2	Solve complex heterogeneous data intensive analytical based problems of real time scenario using state of the art hardware/software tools	1	Laboratory Practices, student viva
PO 5	Independently carry out research/investigation and development work to solve practical problems	2	Laboratory Practices, student viva
PO 7	To engage in life-long learning and professional development through self-study, continuing education, professional and doctoral level studies.	3	Laboratory Practices, Mini project

3 = High; 2 = Medium; 1 = Low

VII. COURSE OBJECTIVES :

The course should enable the students to:	
I	Analyze various sorting techniques.
II	Implement linear and nonlinear data structures and hashing techniques like probing
III	Develop applications of data structures
III	Choose appropriate data structure and algorithm design method for a specific application.
IV	Identify suitable data structure to solve various computing problems.

VIII. COURSE OUTCOMES (COs):

CO Code	CO's	At the end of the course, the student will have the ability to:	PO's Mapped	Strength of Mapping
BCSB09.01	CO 1	Implement divide and conquer techniques to solve a given problem	PO 1	3
BCSB09.02	CO 2	Implement hashing techniques like linear probing, quadratic probing, random probing and double hashing/rehashing	PO 2	3
BCSB09.03	CO 3	Perform Stack operations to convert infix expression into post fix expression and evaluate the post fix expression.	PO 5	3
BCSB09.04	CO 4	Differentiate graph traversal techniques Like DFS,BFS	PO 2	3
BCSB09.05	CO 5	Identify shortest path to other vertices using various algorithms.	PO 5	3

3 = High; 2 = Medium; 1 = Low

IX. MAPPING COURSE LEARNING OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcomes (COs)	Program Outcomes						
	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO 1	3						
CO 2		3					
CO 3					3		
CO 4		3					
CO 5					3		

3 = High; 2 = Medium; 1 = Low

X. ASSESSMENT METHODOLOGIES – DIRECT

CIE Exams	PO 1, PO 2, PO 5,PO 7	SEE Exams	PO 1, PO 2, PO 5 PO 7	Laboratory Practices	PO 1, PO 2, PO 5, PO 7
Mini Project	PO7	Student Viva	PO 1, PO 2, PO 5		

XI. ASSESSMENT METHODOLOGIES - INDIRECT

✔	Early Semester Feedback	✔	End Semester OBE Feedback
✗	Assessment of Mini Projects by Experts		

XII. SYLLABUS

LIST OF EXPERIMENTS	
Week-1	**DIVIDE AND CONQUER - 1**
a. Implement Quick Sort on 1D array of Student structure (contains student name, student_roll_no,total_marks), with key as student_roll_no and count the number of swap performed. b. Implement Merge Sort on 1D array of Student structure (contains student_name, student_roll_no,total_marks), with key as student_roll_no and count the number of swap performed.	
Week-2	**DIVIDE AND CONQUER - 2**
a. Design and analyze a divide and conquer algorithm for following maximum sub-array sum problem:given an array of integer's find a sub-array [a contagious portion of the array] which gives the maximum sum. b. Design a binary search on 1D array of Employee structure (contains employee_name, emp_no, emp_salary), with key as emp_no and count the number of comparison happened.	
Week-3	**IMPLEMENTATION OF STACK AND QUEUE**
a. Implement 3-stacks of size 'm' in an array of size 'n' with all the basic operations such as Is Empty(i),Push(i), Pop(i), IsFull(i) where 'i' denotes the stack number (1,2,3), Stacks are not overlapping each other. b. Design and implement Queue and its operations using Arrays	
Week-4	**HASHING TECHNIQUES**
Write a program to store k keys into an array of size n at the location computed using a hash function, loc =key % n, where k<=n and k takes values from [1 to m], m>n. To handle the collisions use the following collision resolution techniques a. Linear probing b. Quadratic probing c. Random probing d. Double hashing/rehashing	
Week-5	**APPLICATIONS OF STACK**
Write C programs for the following: a. Uses Stack operations to convert infix expression into post fix expression. b. Uses Stack operations for evaluating the post fix expression.	
Week-6	**BINARY SEARCH TREE**
Write a program for Binary Search Tree to implement following operations: a. Insertion b. Deletion i. Delete node with only child ii. Delete node with both children c. Finding an element d. Finding Min element e. Finding Max element f. Left child of the given node g. Right child of the given node h. Finding the number of nodes, leaves nodes, full nodes, ancestors, descendants.	

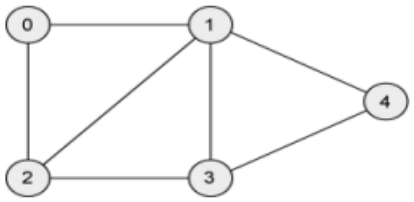
Week-7	DISJOINT SET OPERATIONS

a. Write a program to implement Make_Set, Find_Set and Union functions for Disjoint Set Data Structure for a given undirected graph G(V,E) using the linked list representation with simple implementation of Union operation.

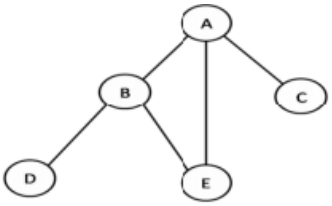
b. Write a program to implement Make_Set, Find_Set and Union functions for Disjoint Set Data Structure for a given undirected graph G(V,E) using the linked list representation with weighted-union heuristic approach

Week-8	GRAPH TRAVERSAL TECHNIQUES

a. Print all the nodes reachable from a given starting node in a digraph using BFS method.

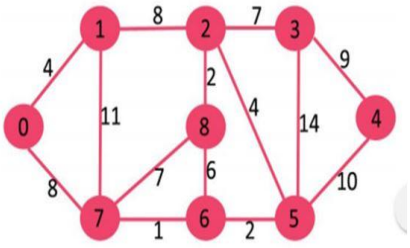


b. Check whether a given graph is connected or not using DFS method.



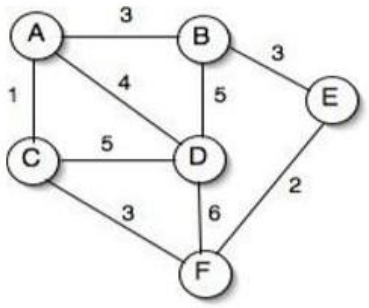
Week-9	SHORTEST PATHS ALGORITHM

From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.



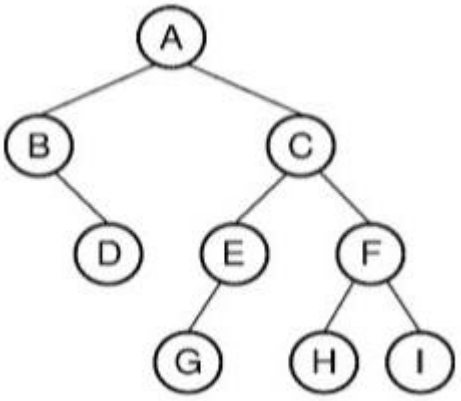
Week-10	MINIMUM COST SPANNING TREE

Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's



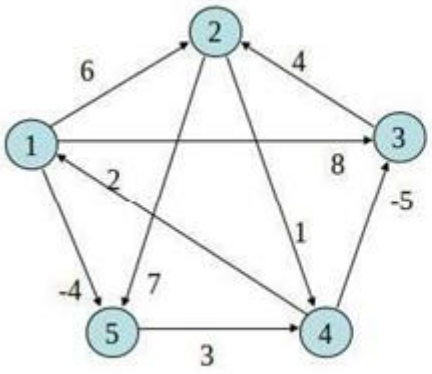
WeeK-11	TREE TRAVESRSALS

Perform various tree traversal algorithms for a given tree.



Week-12	ALL PAIRS SHORTEST PATHS

Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.



	1	2	3	4	5
1	0	6	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	∞	-5	0	∞
5	∞	∞	∞	3	0

Reference Books:

1. Kernighan Brian W, Dennis M. Ritchie, “The C Programming Language”, Prentice Hall of India, RePrint, 2008.
2. Balagurusamy E, “Programming in ANSIC”, Tata McGraw Hill, 6th Edition, 2008.
3. Gottfried Byron, “Schaum's Outline of Programming with C”, Tata McGraw Hill, 1st Edition, 2010.
4. Lipschutz Seymour, “Data Structures Schaum's Outlines Series”, Tata McGraw Hill, 3rdEdition, 2014.
5. Horowitz Ellis, Satraj Sahni, Susan Anderson, Freed, “Fundamentals of Data Structures in C”, W. H.Freeman Company, 2ndEdition, 2011.

Web References:

1. http://www.tutorialspoint.com/data_structures_algorithms
2. http://www.geeksforgeeks.org/data-structures/
3. http://www.studytonight.com/data-structures/
4. http://www.coursera.org/specializations/data-structures-algorithms

SOFTWARE AND HARDWARE REQUIREMENTS FOR A BATCH OF 60 STUDENTS:

HARDWARE:18 numbers of Intel Desktop Computers with 4 GB RAM.
SOFTWARE: Python

XIII. COURSE PLAN:

The course plan is meant as a guideline. Probably there may be changes.

Lecture No.	Learning Objectives	Topics to be covered
1-3	Understand sorting techniques	Quick sort and Merge sort
4-6	Understand divide and conquer techniques	Design and analyze a divide and conquer algorithm
7-9	Implement data structures like stack and queue	Implementation stack and queue and its operations using arrays
10-12	Implement hashing techniques	Understand collision resolution techniques
13-15	Understand applications of stacks	Evaluate postfix expression using stack
16-18	Understand insertion and deletion on binary search trees	Implement traversals on binary search tree
19-21	Understand the implementation of disjoint set operations	Implement Disjoint Set Operations
22-24	Understand traversals on graphs	Implement graph traversal techniques like DFS and BFS
25-27	Implementation of shortest path using Dijkstra's algorithm	Design a shortest paths algorithm using Dijkstra's algorithm
28-30	Implementation of minimum cost spanning tree	Minimum Cost Spanning Tree using Kruskal's algorithm
31-33	Implementation of traversals on trees	Implement Tree Traversals
34-36	Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.	All-Pairs Shortest Paths Problem using Floyd's algorithm.
37-39	Internal Lab Exam	CIE-I

Prepared by:
Ms. S Swarajya Laxmi

HOD, CSE