COURSE CONTENT

ADVANCED DATA STRUCTURES LABORATORY								
I Semester: CSE								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
BCSD11	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		0	0	4	2	40	60	100
Contact Classes: Nil	**Tutorial Classes: Nil**	**Practical Classes: 36**				**Total Classes:36**		

I. COURSE OVERVIEW:

The course covers the selection and application of advanced data structures as tools for algorithm design to solve problems. The topics include binary trees, binary search trees, and graph traversals. The course enables the students to select and apply data structures that are appropriate for problems in problem-solving in engineering areas.

II. COURSE OBJECTIVES:

The students will try to learn:

I. The linear and nonlinear data structures and their implementations.
II. Algorithms analysis based on their time and space complexity.
III. Appropriate data structure and algorithm design method for a specific application.
IV. The graph traversals algorithms to solve real-world challenges such as finding the shortest paths onhuge maps and assembling genomes from millions of pieces.

III. COURSE OUTCOMES:

After successful completion of the course, students will be able to:

CO1	Apply the divide and conquer technique using datastructures and ADT/libraries to solve the problems	Apply
CO2	Use stack operations for evaluating mathematical expressions and implement operations of queues	Understand
CO3	Demonstrate collision resolution techniques with hashingtechnique	Apply
CO4	Implement operations on sets for graph applications	Analyze
CO5	Use graph traversal algorithms to search vertex in modeling data for graph applications.	Apply
CO6	Apply tree algorithms to solve real life problems	Apply

IV. COURSE CONTENT:

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 integers 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 comparisons 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), Is Full(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 postfix expression.
b. Uses Stack operations for evaluating the postfix expression.

Week 6: BINARY SEARCH TREE

Write a program for Binary Search Tree to implement the 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. The left child of the given node
g. Right child of the given node
h. Finding the number of nodes, leaves nodes, full nodes, ancestors, and descendants.

Week 7: DISJOINT SET OPERATIONS

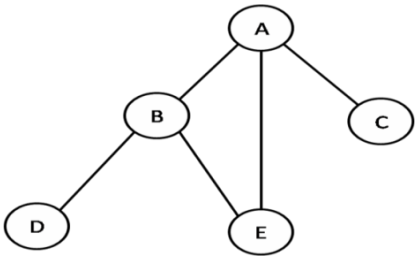
a. Write a program to implement Make_Set, Find_Set and Union functions for Disjoint Set Data Structure for a given undirected graph G(V,E) using the linked list representation with a simple implementation of Union operation.

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 a weighted-union heuristic approach.

Week 8: GRAPH TRAVERSAL TECHNIQUES

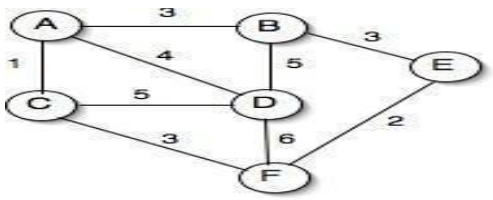
a. Print all the nodes reachable from a given starting node in a digraph using the Breadth First Search (BFS) method.

b. Check whether a given graph is connected or not using the Depth First Search(DFS) method.



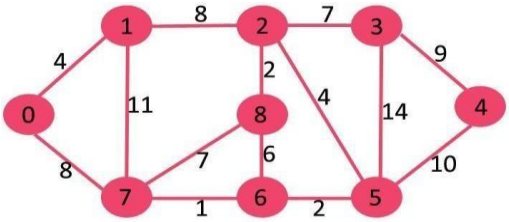
Week-9: SHORTEST PATHS ALGORITHM

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



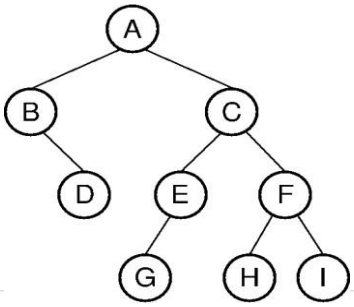
Week 10: MINIMUM COST SPANNING TREE

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



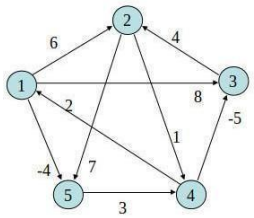
Week 11: TREE TRAVESRSALS

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

V. REFERENCE BOOKS:

1. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, Pearson Education India. Fourth Edition, 2014.
2. Reema Thareja, S. Rama Sree, Advanced Data Structures, Oxford University Press, 2018
3. G.A.V.Pai, "Data Structures and Algorithms", Tata McGraw Hill, NewDelhi,1st Edition,2008
4. Lipschutz Seymour, "Data Structures Schaum's Outlines Series", Tata McGraw Hill, 3rd Edition, 2014.
5. Ellis Horowitz, Sartaj Sahni, Sagathevan Rajasekaran, "Fundamentals of Computer Algorithms", Universities Press Private Limited, India, 2nd Edition, 2008.
6. Adam Drozdek, "Data Structures and Algorithms in C++", Thomson Course Technology, 3rd Edition, 2005

VI. 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