ADVANCE DATA STRUCTURES LABORATORY

I Semester: CSE

Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
BCSB09	Core	-	-	3	2	30	70	100
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 36				Total Classes: 36		

COURSE OBJECTIVES:

The course should enable the students to:

I Implement linear and non linear data structures.

II Analyze various algorithms based on their time complexity.

III Choose appropriate data structure and algorithm design method for a specific application.

IV Identify suitable data structure to solve various computing problems.

COURSE OUTCOMES (COs):

CO 1: Implement divide and conquer techniques to solve a given problem.

CO 2: Implement hashing techniques like linear probing, quadratic probing, random probing and double hashing/rehashing.

CO 3: Perform Stack operations to convert infix expression into post fix expression and evaluate the post fix expression.

CO 4: Differentiate graph traversal techniques Like Depth First Search, Breadth First Search.

CO 5: Identify shortest path to other vertices using various algorithms.

COURSE LEARNING OUTCOMES (CLOs):

1. Analyze time and space complexity of an algorithm for their performance analysis
2. Understand arrays, single and doubly linked lists in linear data structure and tress , graphs in non-linear data structure
3. Master a variety of advanced abstract data type (ADT) and their implementations
4. Understand dynamic data structures and relevant standard algorithms
5. Design and analyze and Concepts of heap, priority queue
6. Analyze probing methods like linear probing and quadratic probing
7. Understand and implement hash table and linear list representation
8. Understand the properties of binary tress and implement recursive and non-recursive traversals
9. Understand graphs terminology, representations and traversals in Graphs
10. Implement Depth First Search and Breath First Searching methods of non –linear data structures
11. Analyze Dijkstra's algorithm for single source shortest path problem for minimum cost spanning trees
12. Implement binary search ADT for finding parent node, smallest and largest values in binary search
13. Understand and implement operations and applications of red-Black and splay Trees
14. Implement Huffman Coding and decoding for text compression.

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:

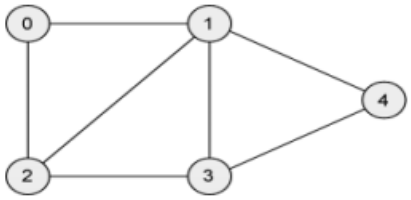
a. Insertion
b. Deletion
   i. Delete node with only child
   ii. Delete node with both children
c. Finding an element
d. Finding Min element
e. Finding Max element
f. Left child of the given node
g. Right child of the given node
h. Finding the number of nodes, leaves nodes, full nodes, ancestors, descendants.

Week-7	DISJOINT SET OPERATIONS

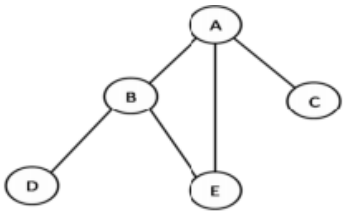
a. Write a program to implement Make_Set, Find_Set and Union functions for Disjoint Set Data Structure for a given undirected graph G(V,E) using the linked list representation with simple implementation of Union operation.
b. Write a program to implement Make_Set, Find_Set and Union functions for Disjoint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

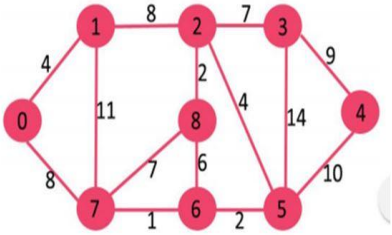


b. Check whether a given graph is connected or not using DFS method.



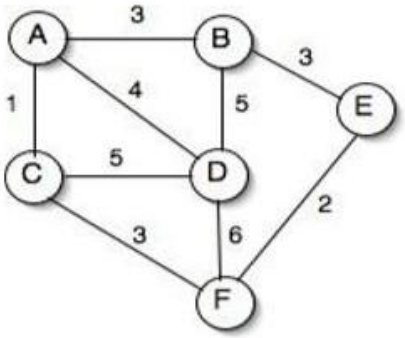
Week-9	SHORTEST PATHS ALGORITHM

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



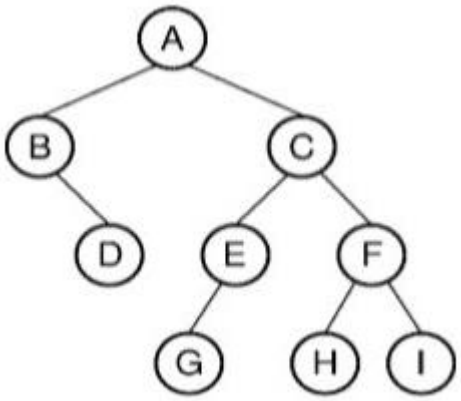
Week-10	MINIMUM COST SPANNING TREE

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



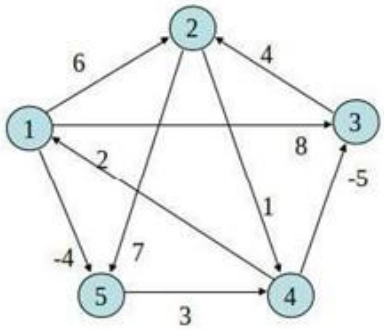
WeeK-11	TREE TRAVESRSALS

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, 3rd Edition, 2014.
5. Horowitz Ellis, Satraj Sahni, Susan Anderson, Freed, "Fundamentals of Data Structures in C", W. H.Freeman Company, 2nd Edition, 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