Register No : _____

THE KAVERY ENGINEERING COLLEGE

(An Autonomous Institution)

B.E/B.TECH DEGREE END SEMESTER THEORY EXAMINATIONS, APRIL/MAY 2026

Third Semester

Computer Science and Engineering

UCST24301 / CS3301 - Data Structures

Regulations – 2024 & 2021

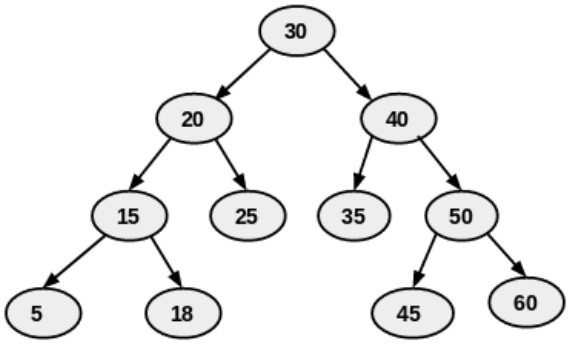
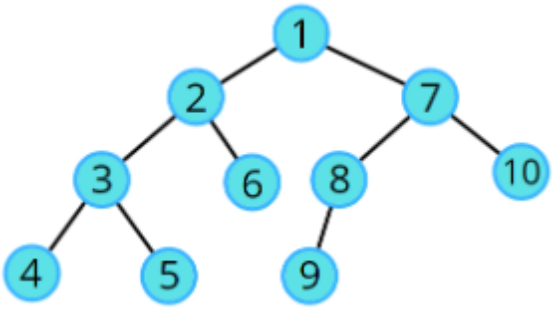
Duration : 3 Hrs *Maximum : 100 Marks*

*PART – A (ANSWER ALL QUESTIONS) (Marks 10*2=20)*

Q.No	*Questions*	*BTL*	*CO*	*Marks*
1.	What is an Abstract Data Type (ADT)?	L1	1	2
2.	Differentiate array-based implementation with linked list implementation.	L2	1	2
3.	Write any two applications of stack.	L1	2	2
4.	What is a circular queue? How is it better than a linear queue?	L2	2	2
5.	What is a binary search tree? List its properties.	L1	3	2
6.	Define binary heap? Distinguish between max heap and min heap.	L2	3	2
7.	Differentiate between directed and undirected graphs.	L2	4	2
8.	Define Euler circuit with an example.	L1	4	2
9.	Compare linear search and binary search.	L2	5	2
10.	State the working principle of selection sort.	L1	5	2

*PART – B (ANSWER ALL QUESTIONS) (Marks 5*16 = 80)*

Q.No			*Questions*	*BLT*	*CO*	*Marks*
11.	a	i	Outline the steps to search an element in a singly linked list with an example and relevant diagrams.	L2	1	8
		ii	Outline the steps to delete an element in a doubly linked list with an example and relevant diagrams	L2	1	8
			Or			
	b	i	Describe how a polynomial is stored using a linked list. Write the algorithm to add two polynomials.	L2	1	8
		ii	Explain Radix Sort algorithm in detail. Provide an example with at least 5 numbers.	L2	1	8
12.	a	i	Explain the stack data structure in detail. Implement it using arrays and perform the following operations in order: push 10, push 20, push 30, pop, push 40, pop, pop. Show the contents of the stack after each operation.	L3	2	8
		ii	Explain how balancing symbols are checked using a stack. Write an algorithm and check whether the following expression is balanced: { [( )] ( ) } and { [( ]) }. Show the state of the stack at each step.	L3	2	8
			Or			

	b	i	Explain infix to postfix conversion using stack. Convert the expression A + B * (C - D) step by step and write the postfix result.	L3	2	8
		ii	Explain the queue data structure in detail. Using an array of size 5, perform enqueue operations for elements 11, 22, 33, 44 and dequeue twice. Provide the algorithms for enqueue and dequeue.	L3	2	8
13.	a	i	Explain tree traversals in detail. Perform preorder, inorder and postorder traversals on the tree given below: 	L3	3	8
		ii	Explain the binary tree ADT. Implement it using linked representation. Insert the following elements in level order and draw the resulting binary tree. 15, 10, 20, 8, 12, 17, 25.	L3	3	8
			Or			
	b	i	Explain the deletion operation in a binary heap. Apply it to the max heap formed by inserting 45, 20, 35, 15, 10.	L3	3	8
		ii	Compare binary search trees and AVL trees in terms of structure, height, and balancing operations. Insert the elements 15, 10, 20, 8, 12, 16, 25 into an AVL tree and perform required rotations in-order to maintain the balance parameter.	L3	3	8
14.	a	i	Explain B-Tree in detail. Insert the following keys into a B-Tree of order 3: 10, 20, 5, 6, 12, 30, 7, 17. Show the tree after each insertion.	L3	4	8
		ii	Explain B+ Tree. Insert the following keys into a B+ Tree of order 3: 15, 25, 35, 45, 55, 65. Show how nodes are split and how leaves are linked.	L3	4	8
			Or			
	b	i	Explain directed, undirected, weighted, unweighted, cyclic and acyclic graphs with suitable example.	L3	4	8
		ii	Explain Breadth-First Traversal (BFS) in graphs. Apply BFS on the graph given below starting from vertex 1. 	L3	4	8

15.	a	i	Explain selection sort and sort the array [29, 10, 14, 37, 13] using selection sort. Write a pseudocode for this sorting.	L3	5	8
		ii	Explain the working principal of merge sort. Sort the numbers 38, 27, 43, 3, 9, 82 and 10 using merge sort.	L3	5	8
			Or			
	b	i	Explain hashing and hash functions. Using a hash table of size 10, insert the following keys [23, 43, 12, 56, 72] using the hash function h(k) = k mod 10. Show the table after insertion.	L3	5	8
		ii	Explain open addressing techniques and rehashing. Using the same set of keys [23, 43, 12, 56, 72], show how collisions are resolved using linear probing and rehashing when the table size is 7.	L3	5	8