U.S.N.										

B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

June 2025 Semester End Main Examinations

Programme: B.E. **Semester: VI**
Branch: Institutional Elective **Duration: 3 hrs.**
Course Code: 23IS6OEDSA / 22IS6OEDSA **Max Marks: 100**
Course: Data Structures and Algorithms

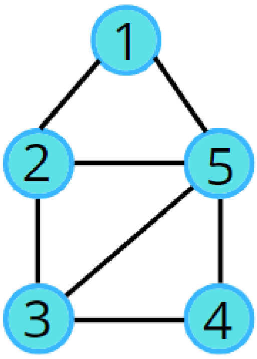
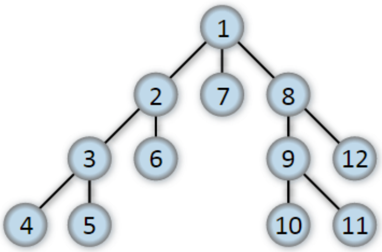
Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

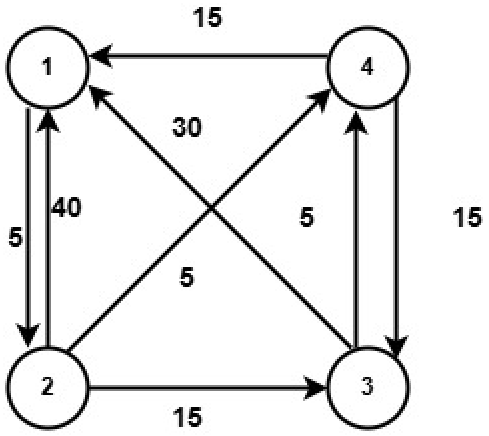
Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

		UNIT - I	***CO***	***PO***	**Marks**
1	a)	Define Data structures. Explain the classification of data structures with examples.	*CO1*	*PO1*	**07**
	b)	Develop a C routine to insert an element to the front of singly linked list.	*CO1*	*PO1*	**08**
	c)	Write a program to find the sum of all the elements in the circular singly linked list.	*CO1*	*PO1*	**05**
		OR			
2	a)	Explain Dynamic memory allocation functions giving syntax and examples.	*CO1*	*PO1*	**06**
	b)	Develop a C routine to insert an element to the end of a doubly linked list.	*CO1*	*PO1*	**08**
	c)	Develop a function to delete a given node in a singly linked list.	*CO1*	*PO1*	**06**
		UNIT - II			
3	a)	Convert the infix into postfix showing the stack contents i) a*(b+d)/e-f*(g+h/k) ii) ((A + B) * C - D) / E	*CO1*	*PO1*	**08**
	b)	i) Given a scenario where person1 is typing a few words in a document: "Hello", then "World", then "Again". When you press **Undo**, the last word disappears. Which data structure helps implement this Undo feature? Write C function for insertion and deletion in that data structure. If you Undo twice, which words will be removed? ii) Write a function to reverse a string using a stack.	*CO2*	*PO2*	**06**
	c)	Develop a program to find factorial of a number using recursion. Illustrate the stack contents.	*CO1*	*PO1*	**06**

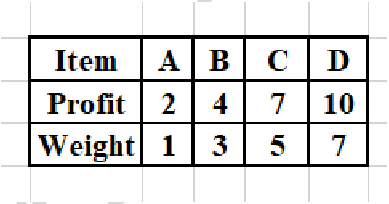
			OR			
	4	a)	Identify the data structure that follows FIFO. Write the functions to insert and delete elements from the identified data structure.	*CO2*	*PO2*	**07**
		b)	Write an algorithm to evaluate a postfix expression. Evaluate the postfix expression 623+-382/+*2*3+.	*CO1*	*PO1*	**08**
		c)	Consider a circular queue implemented using an array of size 5. Initially, the queue is empty. Perform the following operations in sequence: *1.* Insert 10 *2.* Insert 20 *3.* Insert 30 *4.* Delete *5.* Insert 40 *6.* Insert 50 *7.* Insert 60 *8.* Insert 100 Show the queue contents after each operations.After performing these operations, what are the elements present in the queue, and what are the positions of the front and rear pointers?	*CO2*	*PO2*	**05**
			UNIT - III			
	5	a)	Provide the need of an Asymptotic notation? Explain the different Asymptotic notations with definitions and examples.	*CO2*	*PO2*	**06**
		b)	Explain the general plan for analyzing the efficiency of a recursive algorithm. Write the algorithm to find a factorial of a given number. Derive its efficiency.	*CO1*	*PO1*	**08**
		c)	i) Consider the Algorithm **ALGORITHM Sum(n)** **// Input: A nonnegative integer(n)** **S = 0** **for i =1 to n do** **S = S + i** **return S** a. What does this algorithm compute? b. Considering the, basic operation as —S =S + i , Find out how many times the basic operation will be executed? c. What is the efficiency class of this algorithm? ii) a). Find out how many times the statement —printf(“*”); will be executed for the following code given below b). What is the Time Complexity of this code ? **function(int n)** **{** **if (n==1)** **return;** **for (int i=1; i<=n; i++)**	*CO2*	*PO2*	**06**

			{
			for (int j=1; j<=n; j++)
			{
			printf("*");
			break;
			}
			} }			

			OR			
	6	a)	Construct Binary search tree for the given set of values 140,150,40,90,70,180,3,5,165,200. Also perform in order, preorder and postorder traversals of the obtained tree.	***CO1***	***PO1***	**08**
		b)	Write function to insert a element to a Binary search tree.	***CO1***	***PO1***	**08**
		c)	Define the following :i) Complete binary tree ii) Binary Search Tree	***CO1***	***PO1***	**04**
			UNIT - IV			
	7	a)	Array1=[38, 27, 43, 3, 9, 82, 10]. Let Array1 represent score of 7 students in an assessment. Apply merge sort for the array showing the steps.	***CO1***	***PO1***	**08**
		b)	Provide a program to sort the elements in ascending order using selection sort.	***CO1***	***PO1***	**06**
		c)	 Apply BFS algorithm for the given graph. Also show the queue contents during each step.	***CO1***	***PO1***	**06**
			OR			
	8	a)	Write an algorithm to sort the elements using quick sort.	***CO1***	***PO1***	**08**
		b)	 Apply DFS algorithm for the given graph.	***CO1***	***PO1***	**06**

<table>
<tr><td></td><td></td><td>c)</td><td>Consider the heights of 7 friends as 7, 12, 9, 11, 3, Apply bubble sort showing all the intermediate steps to sort in ascending order of heights.</td><td><i>CO2</i></td><td><i>PO2</i></td><td><b>06</b></td></tr>
<tr><td></td><td></td><td></td><td><b>UNIT - V</b></td><td></td><td></td><td></td></tr>
<tr><td></td><td>9</td><td>a)</td><td>Write the algorithm for finding all pairs shortest paths by using Floyds algorithm. Apply the Floyd’s algorithm for the given graph providing all the steps.


15
1
4
30
40
5
5
15
5
2
3
15

</td><td><i>CO1</i></td><td><i>PO1</i></td><td><b>10</b></td></tr>
<tr><td></td><td></td><td>b)</td><td>Apply heap sort for the given elements [10, 3, 76, 34, 23, 32]</td><td><i>CO1</i></td><td><i>PO1</i></td><td><b>10</b></td></tr>
<tr><td></td><td></td><td></td><td><b>OR</b></td><td></td><td></td><td></td></tr>
<tr><td></td><td>10</td><td>a)</td><td>Explain linear and quadratic probing collision avoiding techniques with examples.</td><td><i>CO1</i></td><td><i>PO1</i></td><td><b>10</b></td></tr>
<tr><td></td><td></td><td>b)</td><td>Consider the capacity of the knapsack is W = 8 and the items are as shown in the following table. Solve the problem using dynamic programming.
<table><tr><th>Item</th><th>A</th><th>B</th><th>C</th><th>D</th></tr><tr><td>Profit</td><td>2</td><td>4</td><td>7</td><td>10</td></tr><tr><td>Weight</td><td>1</td><td>3</td><td>5</td><td>7</td></tr></table></td><td><i>CO1</i></td><td><i>PO1</i></td><td><b>10</b></td></tr>
</table>

******