U.S.N.										

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

Autonomous Institute Affiliated to VTU

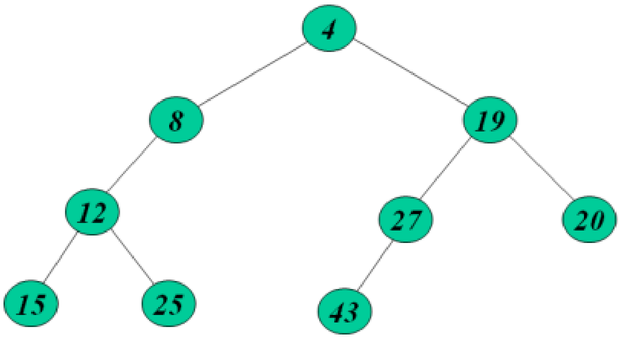
April 2025 Semester End Make-Up Examinations

Programme: B.E.	**Semester: V**
Branch: Information Science and Engineering	**Duration: 3 hrs.**
Course Code: 23IS5PEADS	**Max Marks: 100**
Course: Advanced 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.

		UNIT - I	***CO***	***PO***	**Marks**
1	a)	A B-Tree is used to store product IDs for a large retail database. Each product ID is a key, and the tree is of order 4. Initially, the B-Tree is empty. Insert the following keys in order: 10, 20, 5, 6, 15, 25, 30, 12, 18, 28. i. Show the structure of the B-Tree after all insertions. ii. Describe how the tree handles node splits when inserting 15 and 25.	*CO2*	*PO2*	**9**
	b)	Compare the Interval Trees and Segment Trees.	*CO2*	*PO1*	**5**
	c)	Build a segment tree for the array [2, 4, 1, 3, 6, 5, 8, 7] • Demonstrate the construction process. • Use the tree to answer range sum queries for the ranges [1,4] and [3,7].	*CO1*	*PO2*	**6**
		OR			
2	a)	A hospital uses an interval tree to manage room usage times. Each room's usage is represented by [start, end]. • Construct the interval tree for the intervals [8:00, 9:30], [10:00, 11:30], [9:15, 10:45], [12:00, 1:30]. • Query for any overlapping rooms for the time interval [9:00,10:00].	*CO2*	*PO2*	**8**
	b)	Design a skip list to store and efficiently query a list of userid's {501, 502, 504, 509, 510, 515} in an online gaming platform. Include operations to search, insert, and delete userid's. Level 2 : 502, 510 Level 1 : 501, 509 Level 0 : 504, 515	*CO1*	*PO2*	**8**
	c)	What are the Properties of B-Trees?	*CO2*	*PO1*	**4**

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.

<table>
<tr><td></td><td></td><td></td><td><b>UNIT - II</b></td><td></td><td></td><td></td></tr>
<tr><td></td><td>3</td><td>a)</td><td>Construct a trie for the following words: {"tree", "trie", "trip", "trap", "tap"}. Show the structure and explain how it supports efficient prefix search.</td><td><i>CO2</i></td><td><i>PO2</i></td><td><b>8</b></td></tr>
<tr><td></td><td></td><td>b)</td><td>Perform a union operation on the following two binomial heaps:
• Heap 1: [10,15,25] where $B_0 = 10$
• Heap 2: [5,20,30] where $B_0 = 5$
Show the resulting structure and discuss the time complexity of the operation.</td><td><i>CO1</i></td><td><i>PO9</i></td><td><b>6</b></td></tr>
<tr><td></td><td></td><td>c)</td><td>Compare the usage of leftist heaps and binomial heaps. Discuss their advantages and limitations.</td><td><i>CO1</i></td><td><i>PO1</i></td><td><b>6</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>4</td><td>a)</td><td>For the given tree, identify if it is a leftist heap, Justify the role of the null path length (NPL) in maintaining the heap property.

 </td><td><i>CO3</i></td><td><i>PO2</i></td><td><b>6</b></td></tr>
<tr><td></td><td></td><td>b)</td><td>Write the differences between the tries and suffix trees.</td><td><i>CO3</i></td><td><i>PO1</i></td><td><b>6</b></td></tr>
<tr><td></td><td></td><td>c)</td><td>For the string " abracadabra", construct a suffix tree and use it to count the occurrences of the substring "ab".</td><td><i>CO3</i></td><td><i>PO4</i></td><td><b>8</b></td></tr>
<tr><td></td><td></td><td></td><td><b>UNIT - III</b></td><td></td><td></td><td></td></tr>
<tr><td></td><td>5</td><td>a)</td><td>Create Huffman codes for a scenario where a streaming service compresses data packets with the following character probabilities:
<table><tr><th>Character</th><th>Probability (%)</th></tr><tr><td>A</td><td>30</td></tr><tr><td>B</td><td>25</td></tr><tr><td>C</td><td>20</td></tr><tr><td>D</td><td>15</td></tr><tr><td>E</td><td>10</td></tr></table></td><td><i>CO3</i></td><td><i>PO2</i></td><td><b>8</b></td></tr>
<tr><td></td><td></td><td>b)</td><td>Discuss the time complexity of constructing a Huffman tree using the greedy approach.</td><td><i>CO3</i></td><td><i>PO1</i></td><td><b>4</b></td></tr>
<tr><td></td><td></td><td>c)</td><td>A company processes data using matrix multiplications. The matrices have the following dimensions: A1:10×20, A2:20×30,</td><td><i>CO3</i></td><td><i>PO2</i></td><td><b>8</b></td></tr>
</table>

			A3:30×40, A4:40×50. Determine the optimal parenthesizing to minimize the total number of scalar multiplications.			
			OR			
	6	a)	For the following set of activities: Start | 1 | 3 | 0 | 5 | 8 | 5 | Finish | 2 | 4 | 6 | 7 | 9 | 9 | Show step-by-step how the greedy algorithm selects activities. Compare the result with a brute-force approach.	*CO3*	*PO9*	**8**
		b)	Consider the following keys, probabilities, and depths: • Keys: {Java, Ruby, PHP, Python} • Probabilities: {0.5, 0.3, 0.1, 0.1} • Depths: {2, 1, 2, 3}. Calculate the total search cost of the Optimal Binary Search Tree and verify if the given structure is optimal.	*CO3*	*PO1*	**7**
		c)	Write an exponential-time recursive algorithm to compute the length of an LCS of two sequences.	*CO3*	*PO1*	**5**
			UNIT - IV			
	7	a)	Working modulo q = 13, how many spurious hits does the Rabin-Karp matcher encounter in the text T = 2359023141526739921 when looking for the pattern P = 31415?	*CO4*	*PO5*	**4**
		b)	Construct a transistion diagram for the string-matching automaton that accepts all strings ending in the string “ababaca” and processing the string “aababaababaca” to find the matching or not.	*CO4*	*PO2*	**8**
		c)	Explain the concept of the "prefix function" (or failure function) in the KMP algorithm. How does it help improve efficiency? Illustrate the same.	*CO4*	*PO1*	**8**
			OR			
	8	a)	Construct the string-matching automaton for the pattern P = aabab and illustrate its operation on the text string T = aaababaabaababaab.	*CO4*	*PO2*	**8**
		b)	Write the procedure of Rabin-Karp-Matcher. What is the Worst time complexity of matching.	*CO4*	*PO5*	**7**
		c)	Show how to compute the prefix function for the pattern "AABAACAABAA".	*CO4*	*PO2*	**5**
			UNIT - V			
	9	a)	Given the following graph with vertices A, B, C, D, E and edges: • A→B (4), A→C (−2), B→C (3), C→D (2), D→E (3), E→B (−5). Use the Bellman-Ford algorithm to find the shortest path from A to all other vertices.	*CO5*	*PO10*	**8**
		b)	Write the basic procedure of Ford-Fulkerson algorithm.	*CO5*	*PO1*	**4**
		c)	i. Write the procedure the of Extended-Euclid algorithm.	*CO5*	*PO4*	**8**

			ii. Compute the values (d, x, y) that the call EXTENDED-EUCLID (899, 493) returns.			
			OR			
	10	a)	Design a flow network with 4 nodes. Nodes : S, A, B, T (Source S and sink T) Edges and capacities : S-> A with capacity 10, S-> B with capacity 5, A-> T with capacity 10, B-> T with capacity 5. Define and demonstrate augmenting path, bottleneck capacity, and total flow at each iteration using the Ford-Fulkerson method.	***CO5***	***PO3***	**10**
		b)	A bus arrives at stops every 5, 7 and 9 minutes. If the buses arrive together at 10:00 AM, when will they next arrive together? Solve using Chinese Remainder Theorem.	***CO5***	***PO6***	**10**

******