USN | | | | | | | | | | |

RV COLLEGE OF ENGINEERING®

(An Autonomous Institution affiliated to VTU)

IV Semester B. E. Examinations October-2023

Computer Science and Engineering

DESIGN AND ANALYSIS OF ALGORITHMS

Time: 03 Hours ***Maximum Marks: 100***

Instructions to candidates:

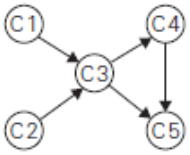
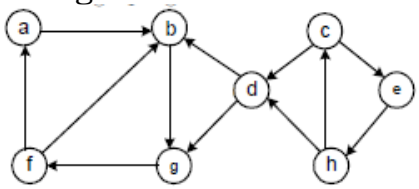
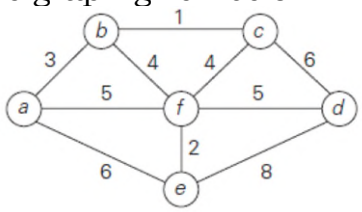
1. Answer all questions from Part A. Part A questions should be answered in first three pages of the answer book only.
2. Answer FIVE full questions from Part B. In Part B question number 2, 7 and 8 are compulsory. Answer any one full question from 3 and 4 & one full question from 5 and 6.

PART-A

1	1.1	State the basic operation in finding factorial of a given number	01
	1.2	Determine the number of times the basic operation gets executed while sorting the following set of elements in ascending order using bubble sort 60 50 40 30 20 10	01
	1.3	Give the time complexity of insertion sort in best case	01
	1.4	Name the design technique used in merge sort algorithm	01
	1.5	What is the asymptotic notation used to denote the tight upper bound?	01
	1.6	What is the time complexity of solving 0/1 knapsack problem using dynamic programming?	01
	1.7	Name the greedy based algorithm used to find shortest path from a given source vertex.	01
	1.8	Name the design technique, which constructs state space tree to find all feasible solutions.	01
	1.9	Calculate the number of character comparisons made by the horspool's algorithm for the pattern $AAAAB$ to be searched in the text 1000 $A's$.	02
	1.10	Apply Master theorem to find the efficiency of the following recurrence: a) $T(n) = 2T(n/2) + n$ b) $T(n) = 3T(n/2) + n^2$	02
	1.11	Name the algorithms used to find the minimum spanning tree of a weighted graph.	02
	1.12	Write the time complexity of Quick Sort in best and worst Case	02
	1.13	Differentiate between DFS and BFS	02
	1.14	Differentiate between backtracking and branch and bound	02

PART-B

2	a	With a neat flowchart, briefly discuss the sequence of steps which typically goes through in designing and analyzing an algorithm	06
	b	Write a Bubblesort algorithm and discuss its efficiency	06
	c	Discuss the general plan for analyzing the Time Efficiency of Recursive Algorithms	04

3	a	Write the pseudocode of quicksort algorithm (along with the partition). Discuss its worst case time complexity.	06
	b	Write the insertion sort algorithm. Discuss its efficiency in the best case	06
	c	Define topological ordering of a graph. Apply *DFS* based algorithm to the given graph to find the topological order.  	04
		OR	
4	a	Write the pseudocode of the merge algorithm used in Mergesort. Apply the same to sort the elements : 8 3 2 9 7 1 5 4	06
	b	Write an algorithm to visit vertices of given graph using DFS and trace the same on the graph given in fig 4b.  Fig 4b	06
	c	Compute 23 × 14 by applying the divide-and-conquer algorithm.	04
5	a	Explain Horspool's algorithm for string matching. Apply the same to search for the following details: Pattern : *BARBER* Text : *JIM_SAW_ME_IN_A_BARBER_SHOP*	06
	b	Write a sorting algorithm using Comparison Counting sort. Apply the same to sort the following elements: 62 31 84 96 19 47	06
	c	Design a presorting-based algorithm to find the element uniqueness in a given set of elements.	04
		OR	
6	a	Write an algorithm to construct a max heap using bottom-up method. Apply the same for the list : 2 9 7 6 5 8	06
	b	Design a presort based algorithm to compute the mode in a given set of elements. Comment on its efficiency.	06
	c	Explain problem reduction with an example to compute *LCM*.	04
7	a	Solve knapsack problem using dynamic Programming for the following instance of the problem having 5 items. Weights(w_i) : 2,1,3,2 Value(v_i) : 12,10,20,15 Capacity(W): 5	06
	b	Write the pseudocode of prim's algorithm to find the minimum spanning tree. Apply the same for the graph given below  Fig 7b.	06

<table>
<tr><td></td><td>c</td><td>Design a dynamic programming based algorithm to compute the binomial coefficient.</td><td>04</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>8</td><td>a</td><td>Write an algorithm to solve the subset-sum problem using backtracking. Apply the same to solve the following instance: $S = \{3, 5, 6, 7\}$ and $d = 15$.</td><td>06</td></tr>
<tr><td></td><td>b</td><td>Solve the following instance of the knapsack problem by the branch-and bound algorithm($W = 16$)
<table>
<tr><th>$item$</th><th>$Weight$</th><th>$value$</th></tr>
<tr><td>1</td><td>4</td><td>$40</td></tr>
<tr><td>2</td><td>7</td><td>$42</td></tr>
<tr><td>3</td><td>5</td><td>$25</td></tr>
<tr><td>4</td><td>3</td><td>$12</td></tr>
</table></td><td>06</td></tr>
<tr><td></td><td>c</td><td>Define P, NP, NP Complete and NP hard problem and establish their relationship with a proper diagram</td><td>04</td></tr>
</table>