16CS43

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

R. V. COLLEGE OF ENGINEERING

Autonomous Institution affiliated to VTU

IV Semester B. E. Examinations April/May-18

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	What is an algorithm?	01
	1.2	Find the average case efficiency of a sequential search algorithm.	01
	1.3	What are the different ways in which graphs can be represented for computer algorithms?	01
	1.4	Define greedy method.	01
	1.5	What is an optimal and feasible solution?	01
	1.6	Denote the time complexity of quick sort algorithm for work case.	01
	1.7	Define binomial coefficient.	01
	1.8	State the principle of backpacking.	01
	1.9	Differentiate between recursive and non-recursive algorithms.	02
	1.10	Write the control abstraction of divide-and-conquer method.	02
	1.11	Compare and contrast DFS and BFS.	02
	1.12	Define the transitive closure of a graph.	02
	1.13	State single source shortest path problem.	02
	1.14	Compare NP-complete and NP-hard problems.	02

PART B

2	a	Discuss with a neat flow chart, the process of algorithm design and analysis.	06
	b	Explain asymptotic notations used in algorithm analysis.	06
	c	If $t_1(u) \in O(g_1(u))$ and $t_2(u) \in O(g_2(u))$, Prove that $t_1(u) + t_2(u) \in O(max\{g_1(u), g_2(u)\})$.	04
3	a	State and discuss to sort a list of elements using merge sort.	04
	b	Discuss how quicksort works to sort an array and trace for the following dataset. $65, 70, 75, 80, 85, 60, 55, 50, 45$. Also draw the tree of recursive calls made.	06
	c	Apply Strassen's algorithm to compute: $\begin{bmatrix} 1 & 0 & 2 & 1 \\ 4 & 1 & 1 & 0 \\ 0 & 1 & 3 & 0 \\ 5 & 0 & 2 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 & 1 \\ 2 & 1 & 0 & 4 \\ 2 & 0 & 1 & 1 \\ 1 & 3 & 5 & 0 \end{bmatrix}$	06

OR

4 a Give a suitable algorithm for finding a minimum edge path between two given vertices in any given graph. Apply that algorithm to the graph shown in Fig Q4(a) showing the tree that identifies the minimum edge path from a to g.

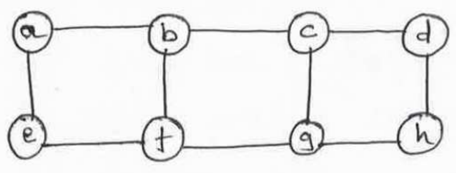

Fig. Q4(a)

04

b Define topological sorting. Apply DFS-based algorithm to solve the topological sorting problem for the given graph Fig Q4(b).

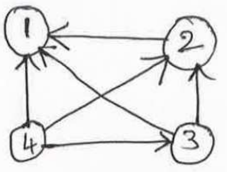

Fig Q4(b)

06

c Write the Johnson Trother algorithm for generating permutations. Generate all permutations of {3,5,7} using the following:

i) Bottom up minimal change algorithm.

ii) Johnson Trother algorithm. 06

5 a Sort the list in non-decreasing order using heep sort. Show the heapification at every step. List: $1, 8, 6, 5, 3, 7, 4$. 06

b Explain the construction of $2 - 3$ tree and construct a $2 - 3$ tree for the list. $10, 6, 9, 4, 3, 5, 8$. 06

c What is AVL tree? Explain the four types of rotations used to construct the AVL tree. 04

OR

6 a State Horspool's algorithm for pattern matching. Apply the same to search for the pattern Brown in a text given below:
Text: $That\_color\_is\_not\_Brown$ 08

b Discuss Boyer-Moore algorithm for string matching. Find the pattern $BAOBAB$ in a given text.
Text: $BESS\_KNEW\_ABOUT\_BAOBAB$ 08

7 a Give the recurrence used to solve Knapsack problem using dynamic programming and explain in brief the same. Solve the following Knapsack problem using dynamic programming. Capacity $w = 5$.

$Item$	1	2	3	4
$Weight$	2	1	3	2
$Value$	12	10	20	15

06

b Write an algorithm to find the all-pairs shortest path problem for the digraph with the weight matrix.

$$\begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \infty & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \infty & 0 \end{bmatrix}$$

04

c Construct a Hauffman code for the following data:

$Char$	A	B	C	D	E
$Probability$	0.1	0.15	0.15	0.2	0.4

Encode the text $BECAD$. 06

8 a With the help of a state space tree, solve the following instance of the Knapsack problem using branch-and-bound technique. Knapsack capacity = 10.

$Item$	1	2	3	4
$Weight$	4	7	5	3
$Value$	40	42	25	12

06

b Explain backtracking concept used to solve Subset_Sum problem. Apply to solve for $S = \{6,5,3,7\}$ and $d = 15$. 06

c Define the following.

i) Class P

ii) Class NP. 04