CBCS SCHEME

USN

Fourth Semester B.E./B.Tech. Degree Examination, June/July 2025
Analysis and Design of Algorithms

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M: Marks , L: Bloom's level , C: Course outcomes.

		Module – 1	M	L	C
Q.1	a.	Define algorithm Explain asymptotic notations Bigh oh, Big omega and Big theta notations.	08	L2	CO1
	b.	Explain the general plan for analyzing the efficiency of a recursive algorithm. Suggest a recursive algorithm to find factorial of number. Derive its efficiency.	08	L3	CO1
	c.	If $t_1(n) \in O(g_1(n))$ and $t_2(n) \in O(g_2(n))$ then show that $t_1(n) + t_2(n) \in o(\max\{g_1(n), g_2(n)\})$	04	L2	CO1
		OR			
Q.2	a.	With a neat diagram explain different steps in designing and analyzing algorithm.	08	L2	CO1
	b.	Write an algorithm to find the max element in an array of n elements. Give the mathematical analysis of this non- recursive algorithm.	08	L3	CO1
	c.	With the algorithm derive the worst case efficiency for selection sort.	04	L3	CO1
		Module – 2			
Q.3	a.	Explain the concept of divide and conquer. Design an algorithm for merge sort and derive its time complexity.	10	L3	CO2
	b.	Design an algorithm for insertion algorithm and obtain its time complexity. Apply insertion sort on these elements. 89, 45, 68, 90, 29, 34, 17	10	L3	CO2
		OR			
Q.4	a.	Design an algorithm for Quick sort. Apply quick sort on these elements. 5, 3, 1, 9, 8, 2, 4, 7.	10	L3	CO2
	b.	Explain Strassen's Matrix multiplication and derive its time complexity.	10	L2	CO2
		Module – 3			
Q.5	a.	Define AVL trees. Explain its four rotation types.	10	L2	CO3
	b.	Design an algorithm for Heap sort. Construct bottom – up heap for the list 15, 19, 10, 7, 17, 16.	10	L3	CO4
		OR			
Q.6	a.	Design Horspool's Algorithm for string matching Apply Horspool algorithm to find pattern BARBER in the test: JIM_SAW_ME_IN_A_BARBERSHOP.	10	L3	CO4
	b.	Define heap. Explain the properties of heap along with its representation.	10	L2	CO3

Module – 4

Q.7 a. Construct minimum cost spanning tree using Kruskal's algorithm for the following graph. (10, L3, CO4)

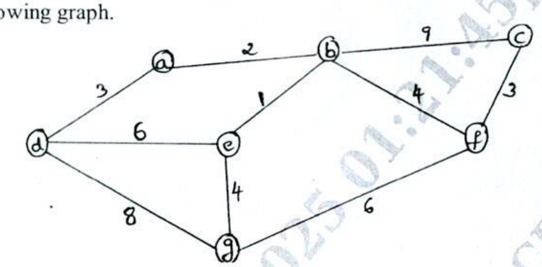


Fig. 7(a)

b. What are Huffman trees? Construct the Huffman tree for the following data (10, L3, CO4)

Character	A	B	C	D	-
Probability	0.4	0.1	0.2	0.15	0.15

i) Encode the text ABAC ABAD

ii) Decode the code100010111001010

OR

Q.8 a. Apply Dijkstra's algorithm to fine single source shortest path for the given graph by considering A as the source vertex. (10, L3, CO4)

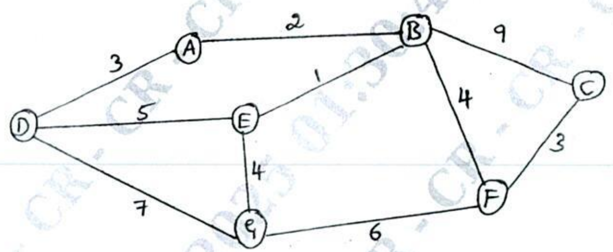


Fig.8 (a)

b. Define transitive closure of a graph. Apply Warshall's algorithm to compute transitive closure of a directed graph. (10, L3, CO4)

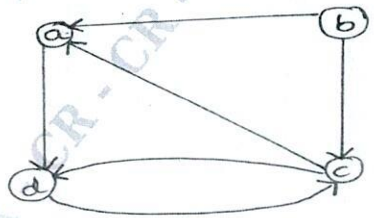


Fig.8 (b)

		Module – 5			
Q.9	a.	Explain the following with examples. i) P problem ii) NP problem ii) NP-Complete problem iv) NP – Hard problem	10	L2	CO5
	b.	What is backtracking? Apply backtracking to solve the below instance of sum of subset problem. S = { 1, 2, 5, 6, 8} and d = 9.	10	L3	CO6
		OR			
Q.10	a.	Illustrate N Queen's problem using backtracking to solve 4 – Queens problem.	10	L2	CO6
	b.	Using Branch and Bound method solve the below instance of Knapsack Problem.	10	L3	CO6

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

Capacity = 10

* * * * *

CBCS SCHEME - Make-Up Exam

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

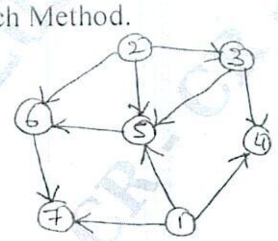
BCS401

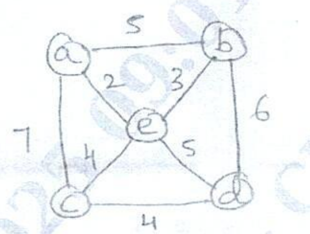
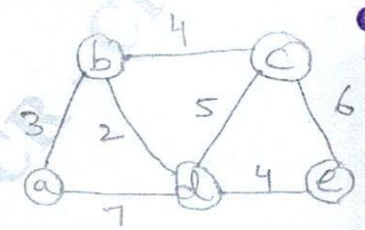
Fourth Semester B.E/B.Tech. Degree Examination, June/July 2025

Analysis & Design of Algorithms

Time: 3 hrs. Max. Marks:100

Note: *1. Answer any FIVE full questions, choosing ONE full question from each module.*
2. M : Marks , L: Bloom's level , C: Course outcomes.

		Module – 1	M	L	C
1	a.	Explain the algorithm design and analysis process in detail.	10	L2	CO1
	b.	Define Algorithm. Explain the asymptotic notations with example.	10	L2	CO1
		OR			
2	a.	Design an algorithm to search an element in an array using sequential search. Discuss the best-case, worst-case and average-case efficiency of this algorithm.	10	L3	CO1
	b.	Give the general plan for analyzing time efficiency of recursive algorithms and also analyze the tower of Hanoi recursive algorithm.	10	L3	CO1
		Module – 2			
3	a.	Apply quick sort algorithm to sort the list : 5, 3, 1, 9, 8, 2, 4, 7. Draw the tree of recursive calls made while tracing.	10	L3	CO2
	b.	Write Merge Sort Algorithm. Find the efficiency of the algorithm.	10	L2	CO2
		OR			
4	a.	Obtain the topological sort for the graph Fig. Q4 (a) using (i) Source Removal Method. (ii) Depth first Search Method.  Fig. Q4 (a)	10	L3	CO2
	b.	Explain Strassen's Matrix Multiplication. Apply Strassen's matrix multiplication to multiply the following matrices: $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$.	10	L3	CO2

		Module – 3			
5	a.	Define AVL tree with an example. Construct an AVL tree of the list of keys : 5, 6, 8, 3, 2, 4, 7 indicating each step of key insertion and rotation.	10	L3	CO3
	b.	Define Heap. Apply heap sort to sort the list of numbers : 2, 9, 7, 6, 5, 8 in ascending order using array representation.	10	L3	CO3
		OR			
6	a.	Define 2-3 Tree. Construct 2-3 tree for the list of keys : 9, 5, 8, 3, 2, 4, 7 by indicating each step of key insertion and node splits.	10	L3	CO3
	b.	Design Har'spool algorithm for string matching. Apply this algorithm to find the pattern BARBER in the text : JIM_SAW_ME_IN_A_BARBERSHOP	10	L3	CO3
		Module – 4			
7	a.	Apply Floyd's algorithm to find the all pair shortest path for the given adjacency matrix. Fig. Q7 (a). $W = \begin{bmatrix} 0 & 1 & \infty & 1 & 5 \\ 9 & 0 & 3 & 2 & \infty \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & \infty & \infty & \infty & 0 \end{bmatrix}$ Fig. Q7 (a)	10	L3	CO4
	b.	Write Kruskal's algorithm to find minimum spanning tree. Illustrate with the following graph Fig. Q7 (b).  Fig. Q7 (b)	10	L3	CO4
		OR			
8	a.	Write Dijkstra's algorithm to find single source shortest path. Apply same for the given Fig. Q8 (a), a as the source vertex.  Fig. Q8 (a)	10	L3	CO4

	b.	Construct a Huffman tree and resulting code word for the following : Symbol: A, B, C, D, - Frequency: 0.35, 0.1, 0.2, 0.2, 0.15 Encode the text DAD. Decode the text 100110110111101.	10	L3	CO4
		Module – 5			
9	a.	Construct a state space tree to solve force queen's problem using backtracking.	10	L3	CO5
	b.	Solve the following instance of the knapsack problem by using branch and bound method. Item: 1, 2, 3, 4 Weight: 4, 7, 5, 3 Value: \$40, \$42, \$25, \$12 Capacity W = 10	10	L3	CO5
		OR			
10	a.	Construct a state space tree for subset sum problem using branch and bound for the set S = {3, 5, 6, 7} and d = 15	10	L3	CO5
	b.	Explain the following terms : (i) P – Problems (ii) NP – Problems (iii) NP – Complete Problems (iv) NP – Hard Problems.	10	L2	CO5

Huffman table (Q8 b):

Symbol	A	B	C	D	-
Frequency	0.35	0.1	0.2	0.2	0.15

Knapsack table (Q9 b):

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

* * * * *

CBCS SCHEME

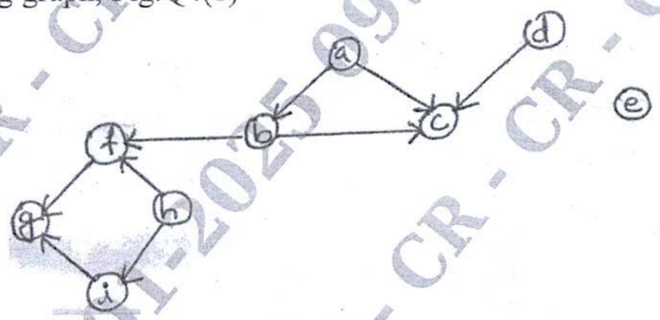
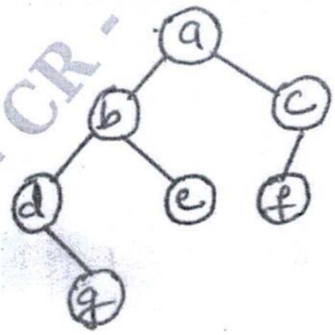
USN | | | | | | | | | | |

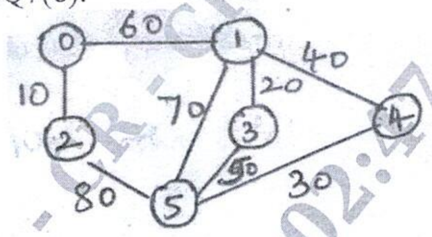
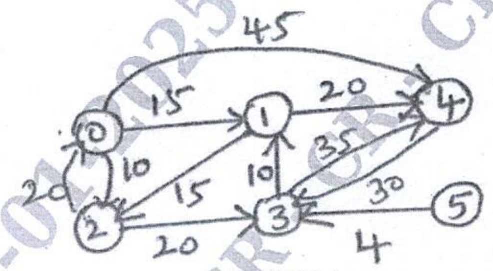
BCS401

Fourth Semester B.E./B.Tech. Degree Examination, Dec.2024/Jan.2025
Analysis and Design of Algorithms

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

		Module – 1	M	L	C
Q.1	a.	Explain the various steps in algorithm design and analysis process with the flow diagram.	08	L1	CO1
	b.	Give formal and informal definitions of asymptotic notations.	06	L1	CO1
	c.	Explain the general plan of mathematical analysis of recursive algorithm with an example.	06	L1	CO1
		OR			
Q.2	a.	Design algorithm for tower of Hanoi problem and obtain time complexity.	10	L1	CO1
	b.	Write an algorithm to search an element in an array using sequential search. Discuss the best case, worst case and average case efficiency of this algorithm.	10	L1	CO1
		Module – 2			
Q.3	a.	Write an algorithm to sort the numbers using insertion sort. Discuss its efficiency.	10	L2	CO2
	b.	Design quick sort algorithm and obtain its best, average and worst case efficiency.	10	L2	CO2
		OR			
Q.4	a.	Write merge sort algorithm and sort the list E X A M P L E.	08	L2	CO2
	b.	Apply the DFS based algorithm to solve the topological sorting problem for the following graph, Fig.Q4(b)  Fig.Q4(b)	06	L3	CO2
	c.	Write algorithm for pre-order, post order and in order traversals of a tree. Write pre-order, in-order and post order for the given tree.  Fig.Q4(c)	06	L2	CO2

		Module – 3			
Q.5	**a.**	Define AVL tree. Construct AVL tree for the list 5, 6, 8, 3, 2, 4, 7.	10	L3	CO3
	b.	Define heap. Sort the following lists by heapsort: H E A P S O R T (in alphabetical order)	10	L3	CO3
		OR			
Q.6	**a.**	Write the algorithm for comparison counting sort. Discuss its efficiency.	10	L2	CO4
	b.	Design Horspools algorithm for string matching. Apply Horspools algorithm to find the pattern BARBER on the text JIM_SAW_ME_IN_BARBERSHOP	10	L3	CO4
		Module – 4			
Q.7	**a.**	Write Warshall's algorithm and apply the same to compute transitive closure of a directed graph. $\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$	10	L3	CO3
	b.	Construct minimum cost spanning tree using Kruskal's algorithm for the following graph, Fig.Q7(b).  Fig.Q7(b)	10	L3	CO4
		OR			
Q.8	**a.**	Solve the following single source shortest path problem assuming vertex '5' as the source.  Fig.Q8(a)	10	L3	CO4
	b.	Write Huffman's algorithm. Construct Huffman tree and resulting code word for the following:	10	L4	CO4

Character	A	B	C	D	E	-
Probability	0.5	0.35	0.5	0.1	0.4	0.2

Encode the text DAD_CBE.

		Module – 5			
Q.9	**a.**	Explain the following with example: (i) P problem (ii) NP problem	06	L1	CO5
	b.	What is decision tree? Construct decision tree for the three element insertion sort.	08	L2	CO5
	c.	Construct state space tree to solve 4 queens problem.	06	L3	CO5

		OR			
Q.10	**a.**	What is backtracking? Apply back tracking to solve the below instance of sum of subset problem: s = {3, 5, 6, 7}, d = 15	**10**	**L3**	**CO6**
	b.	Solve the following instance of knapsack problem using branch and bound technique knapsack capacity = 10.	**10**	**L4**	**CO6**

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

* * * * *

CBCS SCHEME

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

BCS401

Fourth Semester B.E./B.Tech. Degree Examination, June/July 2024
Analysis and Design of Algorithms

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

		Module – 1	M	L	C
Q.1	a.	What is an algorithm? Explain the fundamentals of algorithmic problem solving.	10	L2	CO1
	b.	Develop an algorithm to search an element in an array using sequential search. Calculate the best case, worst case and average case efficiency of this algorithm.	10	L3	CO1
		OR			
Q.2	a.	Explain asymptotic notations with example.	10	L2	CO1
	b.	Give the general plan for analyzing the efficiency of the recursive algorithm. Develop recursive algorithm for computing factorial of a positive number. Calculate the efficiency in terms of order of growth.	10	L3	CO1
		Module – 2			
Q.3	a.	Explain Strassen's matrix multiplication approach with example and derive its time complexity.	10	L3	CO2
	b.	What is divide and conquer? Develop the quick sort algorithm and write its best case. Make use of this algorithm to sort the list of characters: E, X, A, M, P, L, E.	10	L2	CO2
		OR			
Q.4	a.	Distinguish between decrease & conquer and divide & conquer algorithm design techniques with block diagram. Develop insertion sort algorithm to sort a list of integers and estimate the efficiency.	10	L3	CO2
	b.	Define topological sorting. List the two approaches of topological sorting and illustrate with examples.	10	L2	CO2
		Module – 3			
Q.5	a.	Define AVL tree with an example. Give worst case efficiency of operations on AVL tree. Construct an AVL tree of the list of keys: 5, 6, 8, 3, 2, 4, 7 indicating each step of key insertion and rotation.	10	L3	CO3
	b.	Define Heap. Explain the bottom-up heap construction algorithm. Apply heap sort to sort the list of numbers 2, 9, 7, 6, 5, 8 in ascending order using array representation.	10	L3	CO3
		OR			
Q.6	a.	Define 2-3 tree. Give the worst case efficiency of operations on 2-3 tree. Build 2-3 tree for the list of keys 9, 5, 8, 3, 2, 4, 7 by indicating each step of key insertion and node splits.	10	L3	CO3
	b.	Design Horspool algorithm for string matching. Apply this algorithm to find the pattern BARBER in the text: JIM_SAW_ME_IN_A_BARBERSHOP	10	L3	CO3
		Module – 4			
Q.7	a.	Apply Dijkstra's algorithm to find the single source shortest path for given graph [Fig.Q7(a)] by considering 's' as source vertex. Illustrate each step.	10	L3	CO4

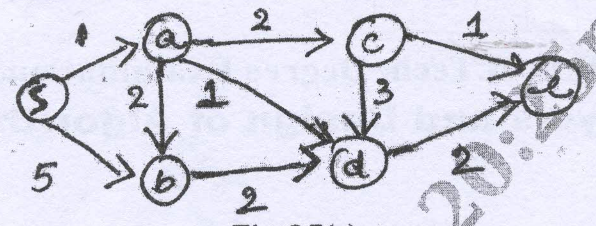


Fig.Q7(a)

b. Define transitive closure. Write Warshall's algorithm to compute transitive closure. Illustrate using the following directed graph. — 10 | L3 | CO4

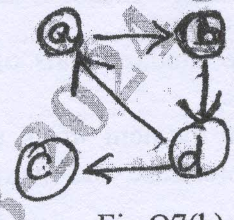


Fig.Q7(b)

OR

Q.8 a. Define minimum spanning tree. Write Kruskal's algorithm to find minimum spanning tree. Illustrate with the following undirected graph. — 10 | L3 | CO4

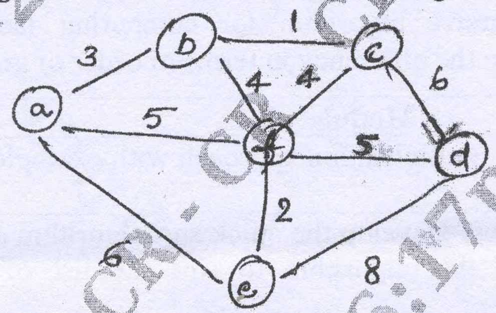


Fig.Q8(a)

b. Construct Huffman Tree and resulting code for the following: — 10 | L3 | CO4

Character	A	B	C	D	-
Probability	0.4	0.1	0.2	0.15	0.15

(i) Encode the text : ABACABAD

(ii) Decode the text : 100010111001010

Module – 5

Q.9 a. Explain n-Queen's problem with example using backtracking approach. — 10 | L2 | CO5

b. Solve the following instance of the knapsack problem by the branch-and-bound algorithm. Construct state-space tree. — 10 | L3 | CO5

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

The knapsack's capacity W is 10.

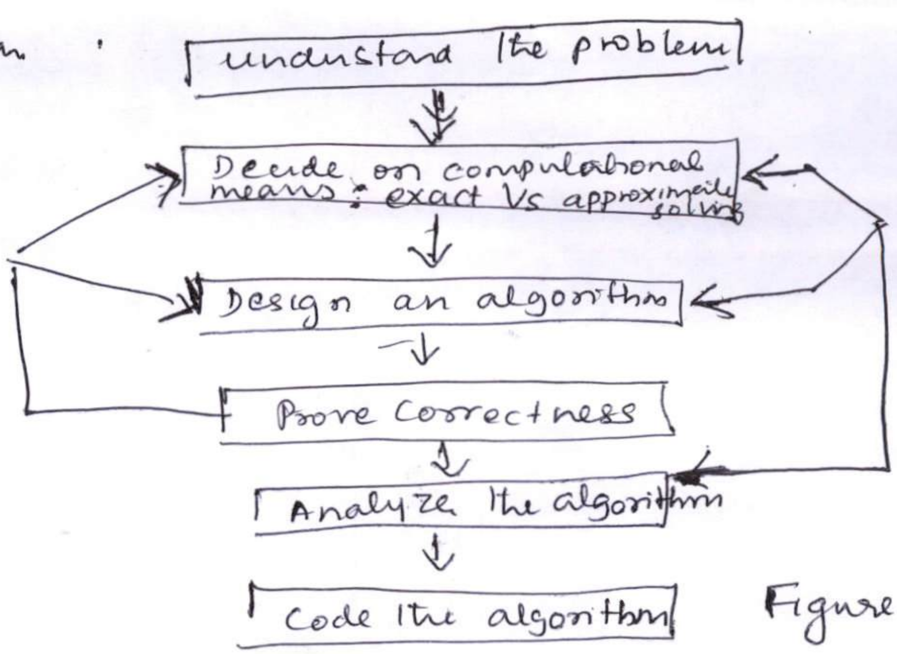
OR

Q.10 a. Differentiate between Branch and Bound technique and Backtracking. Apply backtracking to solve the following instance of subset-sum problem S = {3, 5, 6, 7} and d = 15. Construct a state space tree. — 10 | L3 | CO5

b. Explain greedy approximation algorithm to solve discrete knapsack problem. — 10 | L2 | CO5

*****

Subject Title : Analysis & Design of Algorithms Subject Code : BCS401

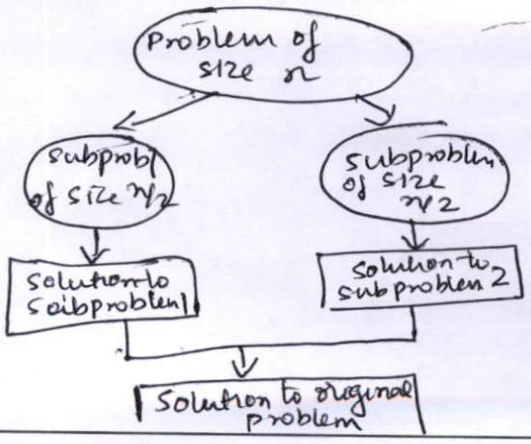
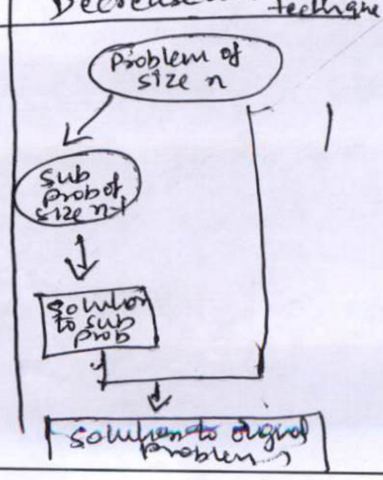
Question Number	Solution	Marks Allocated
1.a.	Algorithm Definition →	1M
	Diagram :  Figure →	3M
	explanation of each step →	6M
	total →	10M
1.b.	Sequential Search Algorithm →	4M
	efficiencies - best case $C_{best}(n) = 1$ →	1M
	worst case $C_{worst}(n) = n$ →	2M
	average case efficiency ~~average case~~ $C_{avg}(n) = [1 \cdot P/n + 2 \cdot P/n + \dots + n\, P/n] + n(1-p)$ $= P/n\,[1 + 2 + \dots + n] + n(1-p)$	

Question Number	Solution	Marks Allocated
	$= \frac{P}{n} \frac{n(n+1)}{2} + n(1-P) = \frac{P(n+1)}{2} + n(1-P)$	1M
	for successful search	
	if $P = 1$ ~~Cavg $\frac{(n+1)}{2}$~~ $C_{avg}(n) = \frac{n+1}{2} \rightarrow$	1M
	for unsuccessful search	
	if $P = 0$	
	$C_{avg}(n) = \underline{n} \longrightarrow$	1M
	total	10M
2.a.	Asymptotic notations:	
	Big-oh (O) → definition, figure & examples →	3M
	Big-omega (Ω) - definitions, figure examples →	3M
	Big-theta (θ) - definition, figure & example	4M
	total →	10M
2.b.	General plans listing: →	
	1. Decide on a parameter indicating an input size. 2. Identify the algorithm's basic operation 3. Check whether the no. of times the basic operation executed can vary on different i/ps of same size.	2M
	4. Set up a Recurrence relation with an 5. initial condition. 5. Solve the recurrence, ascertain the order of growth	1M

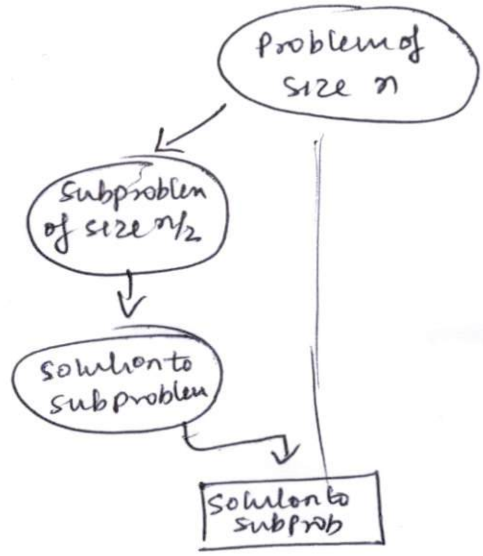
Subject Title : Analysis & Design of Algorithms Subject Code : BCS401

Question Number	Solution	Marks Allocated
	~~Alg.~~ Recursive algorithm for computing factorial number	
	Algorithm F(n) if n=0 return 1 else return F(n-1)* n	2M
	Recurrence relation : $M(n) = M(n-1)+1$ for $n>0$ $M(0) = 0$	2M
	Solving recurrence relation using bwd substitutions	3M
3.a.	Strassen's Matrix Multiplication formula: .	
	$$\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} * \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix}$$ $$= \begin{bmatrix} m_1+m_4-m_5+m_7 & m_3+m_5 \\ m_2+m_4 & m_1+m_3-m_2+m_6 \end{bmatrix}$$ where $m_1 = (a_{00}+a_{11}) * (b_{00}+b_{11})$ $m_2 = (a_{10}+a_{11}) \times b_{00}$ $m_3 = a_{00} \times (b_{01}-b_{11})$ $m_4 = a_{11} \times (b_{10}-b_{00})$ $m_5 = (a_{00}+a_{01}) \times b_{11}$ $m_6 = (a_{10}-a_{00}) \times (b_{00}+b_{01})$ $m_7 = (a_{01}-a_{11}) \times (b_{10}+b_{11})$	3M

Subject Title : Analysis & Design of Algorithms **Subject Code :** BCS401

Question Number	Solution	Marks Allocated
	explanation ⟶	2M
	Recurrence relation of strassen's matrix multiplication $M(n) = 7M(n/2)$ for $n>1$, $M(1) = 1$	2M
	derivation with solution $M(n) = 7^{\log_2 n} = n^{\log_2 7} \approx n^{2.807}$	3M
3.b.	Definition of Divide and Conquer ⟶	1M
	quicksort algo: ⟶	4M
	Sorting the list of characters ⟶	4M
	Quicksort best case efficiency ⟶ $\Theta(n\log n)$	1M
	Total ⟶	10M
4.a.	Divide and Conquer technique : Decrease by one and conquer technique	2M
	 2M 	

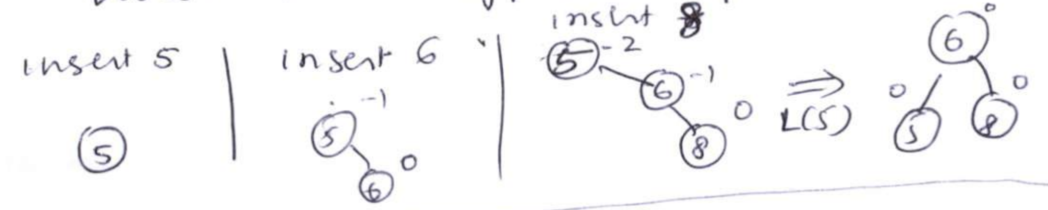
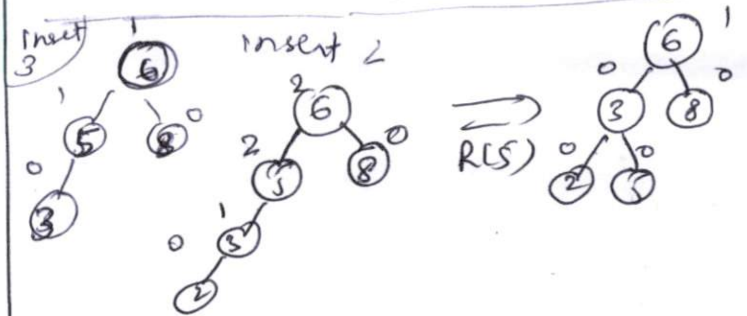
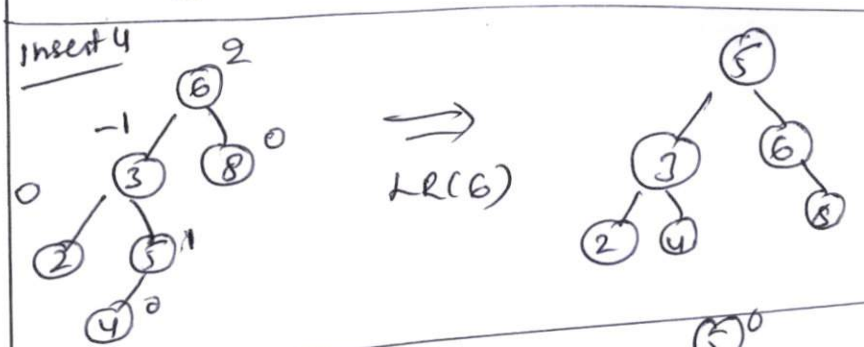
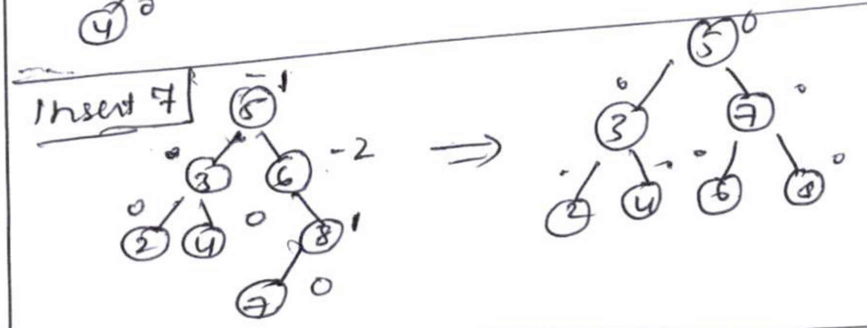
Subject Title : Analysis & Design of Algorithms Subject Code : BCS401

Question Number	Solution	Marks Allocated
	Decrease by half conquer technique:	2M
		2M
	Insertion sort algorithm →	4M

	for $i \leftarrow 1$ to $n-1$ do
	    $v \leftarrow A[i]$
	    $j \leftarrow i-1$
	    while $j \geq 0$ and $A[j] > v$ do
	        $A[j+1] \leftarrow A[j]$
	        $j \leftarrow j-1$
	    $A[j+1] \leftarrow v$	

	$C_{worst}(n) = \Theta(n^2)$ $C_{best}(n) = \Theta(n)$ $C_{avg}(n) = \frac{n^2}{4} \in \Theta(n^2)$	3M
4.b.	Definition of topological sorting : →	1M
	~~List~~ 2 approaches → 1) Source removal 2) DFS based	2M
	Illustration with example (2 approaches)	7M

Subject Title : Analysis & Design of Algorithms Subject Code : BCS401

Question Number	Solution	Marks Allocated
5.a.	AVL definition with example →	2M
	worst case efficiency → Θ(log n)	1M
		1M
		2M
		2M
		2M
	total	10
5.b.	Heap defination & ~~example~~ →	1M
	bottomup construction algorithm (next page) →	3M
	2, 9, 7, 6, 5, 8 → heap construction (in array rep)	3M
	Sorting the numbers (array rep) →	3M

Subject Title : Analysis & Design of Algorithms   Subject Code : BCS401

Question Number	Solution	Marks Allocated
5.b cont.	Bottom up heap construction algorithm:	

for $i \leftarrow \lfloor n/2 \rfloor$ downto 1 do
    $k \leftarrow i$ $v \leftarrow H[k]$
    heap $\leftarrow$ false
while not heap and $2*k \le n$
    $j \leftarrow 2*k$
    if $j < n$
        if $H[j] < H[j+1]$ $j \leftarrow j+1$
    if $v \ge H[j]$
        heap $\leftarrow$ true
    else $H[k] \leftarrow H[j]$; $k \leftarrow j$
$H[k] \leftarrow v$

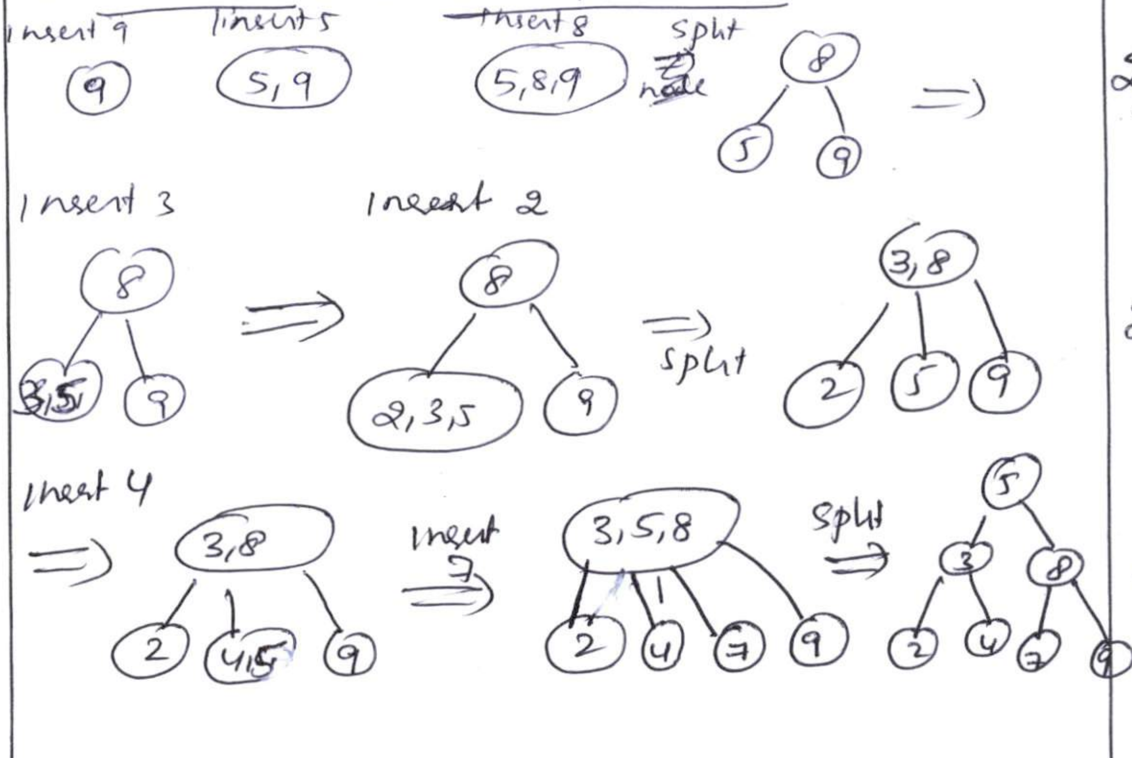
10

6.a

Definition 2-3 tree ~~& example~~ → 2M

worst case efficiency → 1M

construction 2-3 tree 9, 5, 8, 3, 2, 4, 7



2M

2M

3M

Subject Title : Analyzing Design of Algorithms Subject Code : BCS401

Question Number	Solution	Marks Allocated
6.b.	Horspool's algorithm	4M
	Shift table for BARBER	2M

A	B	C	D	E	F	- - - -	R	. . . .	Z	_
4	2	6	6	1	6	6	3	6 . .	6	6

actual search

T: JIM_SAW_ME_IN_A_BARBERSHOP

P: BARBER

BARBER

BARBER

BARBER

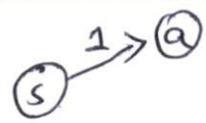
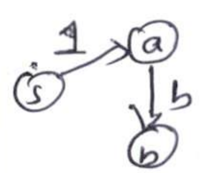
BARBER

BARBER

matched.

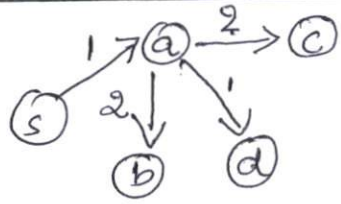
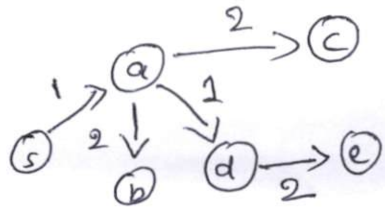
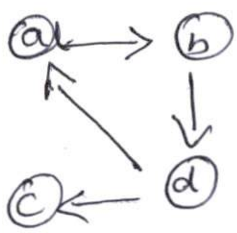
Marks: ~~4M~~ 8 / 4M

7.a.

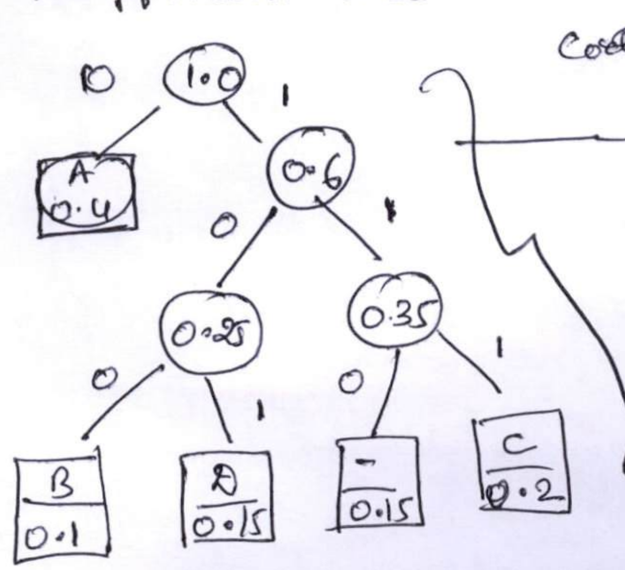
Tree vertices	Remaining Vertices	Illustrate	Marks
s(-,0)	a(s,1) b(s,5) c(-,∞) d(-,∞) e(-,∞)		1M ~~4M~~
a(s,1)	b(a,3) c(a,3) d(a,2) e(-,∞)		2M
~~b(a,3)~~	(b or c) any one can be selected to proceed as both are having same distance		
b(a,3)	c(a,3) d(a,2) e(-,∞)		2M

Cont...next page.

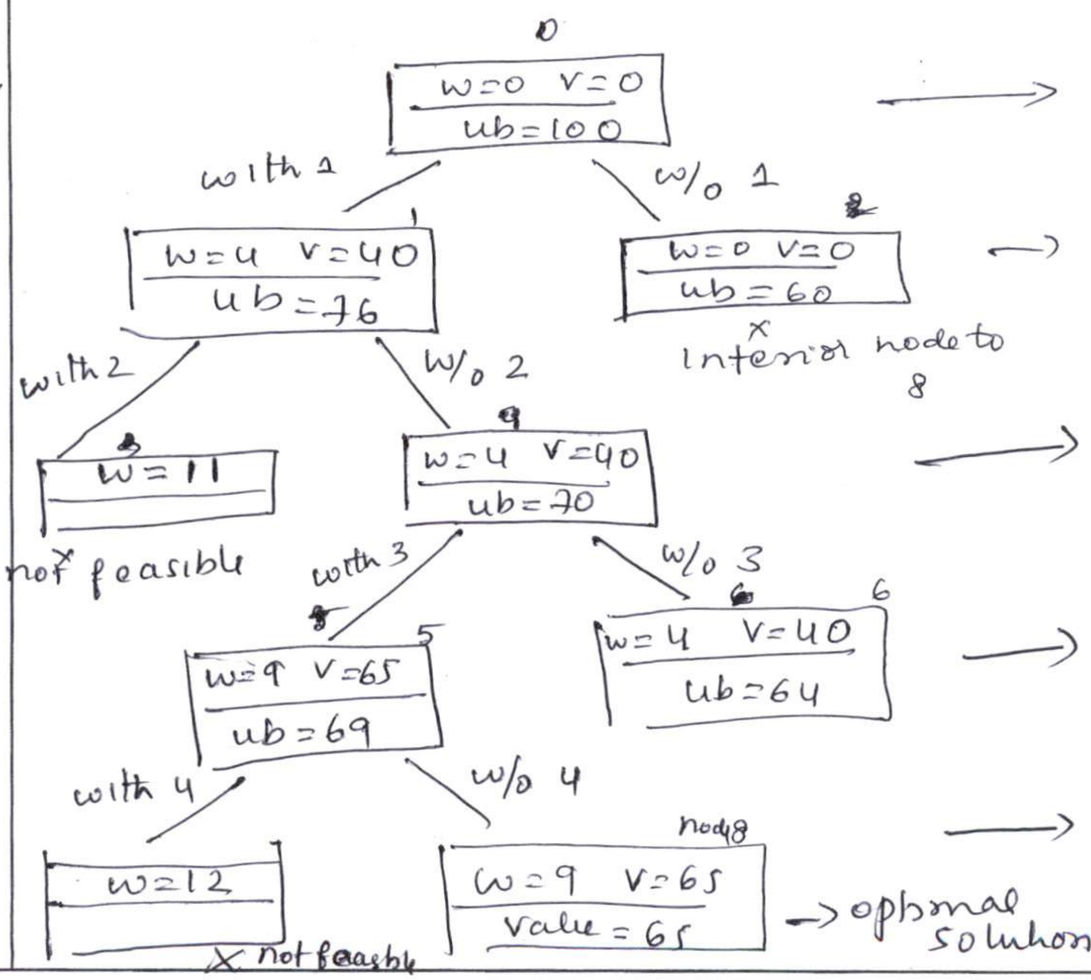
Subject Title : Analysis & Design of Algorithms **Subject Code :** BCS401

Question Number	Solution	Marks Allocated
	d(a,2) | c(a,3) e(d,4) 	2M
	c(a,3) | e(d,4) 	2M
	e(d,4)	1M
	Shortest path $s \to a$ is 1 $s \to a - b$ is 3 $s \to a \to c$ is 3 $s \to a - d$ is 2 $s \to a \to d \to e$ is 4	1M
7.b.	Definition of transitive closure	1M
	~~transit~~ Warshall's algorithm	3M
	 $\to$ Adjacency matrix $\begin{bmatrix} 0&1&0&0 \\ 0&0&0&1 \\ 0&0&0&0 \\ 1&0&1&0 \end{bmatrix}$	1M
	$R^{(0)} = \begin{bmatrix} 0&1&0&0 \\ 0&0&0&1 \\ 0&0&0&0 \\ 1&0&1&0 \end{bmatrix}$ $R^{(1)} = \begin{bmatrix} 0&1&0&0 \\ 0&0&0&1 \\ 0&0&0&0 \\ 1&1&1&0 \end{bmatrix}$	2M
	$R^{(2)} = \begin{bmatrix} 0&1&0&1 \\ 0&0&0&1 \\ 0&0&0&0 \\ 1&1&1&1 \end{bmatrix}$ $R^{(3)} = \begin{bmatrix} 0&1&0&1 \\ 0&0&0&1 \\ 0&0&0&0 \\ 1&1&1&1 \end{bmatrix}$	2M

Subject Title : Analysis & Design of Algorithms **Subject Code :** BCS401

Question Number	Solution	Marks Allocated
	$R^{(4)} : \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$	2M
	$R^{(4)}$ is the transitive closure.	
	~~Efficiency is $O(n^3)$~~	
8.a.	Definition of MST ⟶	1M
	algorithm ⟶	3M
	Steps to derive MST 5 steps ⟶ (bc, ef, ab, bf, df edges)	6M
8.b.	~~Final~~ Stepwise presentation of Huffman tree	
	 Code ⟶	4M
	A → 0 B → 100 C → 111 D → 101 _ → 110	2M

Subject Title : Analysis & Design of Algorithms **Subject Code :** BCSA01

Question Number	Solution	Marks Allocated
	~~Avg # of bits per symbol = 2.2 bits~~	~~1M~~
	~~fixed length encoding require = 3 bits~~	~~1M~~
	~~compression ratio = $100\left(\frac{3-2.2}{3}\right)\times 1$~~	~~2M~~
	~~$= \left(\frac{3-2.2}{3}\right)\times 100\,\%$~~	
	~~$= 26.67\,\%$~~	
	Encode text →	2M
	Decode text →	2M
9.a.	n-queen's problem explanation —	4M
	example with statespace tree —	6M
9.b.		2M 3M 2M 2M 1M

Subject Title : Analysis and design of Algorithms Subject Code : BCS401

Question Number	Solution	Marks Allocated
10.a	Difference → Any 2 × 1M	2M
	State space tree for subset sum problem →	8M
10.b	discrete knapsack problem Algorithm	4M
	Explanation	6M

Model Question Paper with effect from 2023-24 (CBCS Scheme)

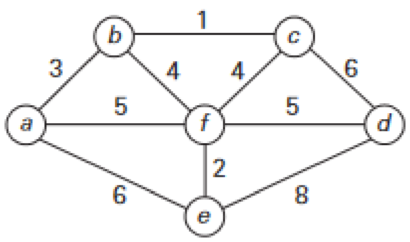
USN | | | | | | | | | |

Fourth Semester B.E. Degree Examination

Analysis and Designs of Algorithms

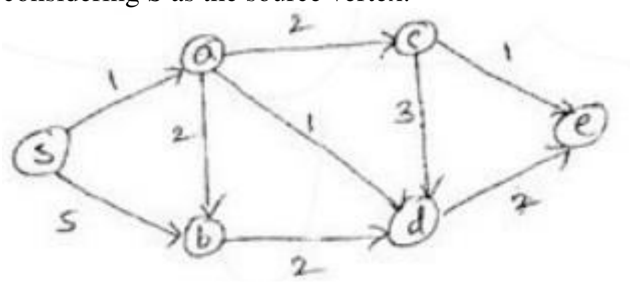
TIME: 03 Hours **Max. Marks: 100**

Note: 01. Answer any **FIVE** full questions, choosing at least **ONE** question from each **MODULE**.

<table>
<tr><th colspan="3">Module -1</th><th>BL</th><th></th><th>Marks</th></tr>
<tr><td>Q.01</td><td>a</td><td>Define algorithm. Explain asymptotic notations Big Oh, Big Omega and Big Theta notations</td><td>L2</td><td></td><td>08</td></tr>
<tr><td></td><td>b</td><td>Explain the general plan for analyzing the efficiency of a recursive algorithm. Suggest a recursive algorithm to find factorial of number. Derive its efficiency</td><td>L2</td><td></td><td>08</td></tr>
<tr><td></td><td>c</td><td>If $t1(n) \in O(g1(n))$ and $t2(n) \in O(g2(n))$, then show that $t1(n) + t2(n) \in O(\max\{g1(n), g2(n)\})$.</td><td>L2</td><td></td><td>04</td></tr>
<tr><td colspan="3">OR</td><td></td><td></td><td></td></tr>
<tr><td>Q.02</td><td>a</td><td>With neat diagram explain different steps in designing and analyzing an algorithm</td><td>L2</td><td></td><td>08</td></tr>
<tr><td></td><td>b</td><td>Explain the general plan for analyzing the efficiency of a non-recursive algorithm. Suggest a non-recursive algorithm to find maximum element in the list of n numbers. Derive its efficiency</td><td>L2</td><td></td><td>08</td></tr>
<tr><td></td><td>c</td><td>With the algorithm derive the worst case efficiency for Bubble sort</td><td>L2</td><td></td><td>04</td></tr>
<tr><td colspan="3">Module-2</td><td></td><td></td><td></td></tr>
<tr><td>Q. 03</td><td>a</td><td>Explain the concept of divide and conquer. Design an algorithm for merge sort and derive its time complexity</td><td>L2</td><td></td><td>10</td></tr>
<tr><td></td><td>b</td><td>Design an insertion sort algorithm and obtain its time complexity. Apply insertion sort on these elements. 25,75,40,10,20,</td><td>L3</td><td></td><td>10</td></tr>
<tr><td colspan="3">OR</td><td></td><td></td><td></td></tr>
<tr><td>Q.04</td><td>a</td><td>Explain Strassen's matrix multiplication and derive its time complexity</td><td>L2</td><td></td><td>10</td></tr>
<tr><td></td><td>b</td><td>Design an algorithm for quick sort algorithm. Apply quick sort on these elements. 25,75,40,10,20,05,15</td><td>L3</td><td></td><td>10</td></tr>
<tr><td colspan="3">Module-3</td><td></td><td></td><td></td></tr>
<tr><td>Q. 05</td><td>a</td><td>Define AVL Trees. Explain its four rotation types</td><td>L2</td><td></td><td>10</td></tr>
<tr><td></td><td>b</td><td>Construct bottom up heap for the list 2,9,7,6,5,8. Obtain its time complexity</td><td>L3</td><td></td><td>10</td></tr>
<tr><td colspan="3">OR</td><td></td><td></td><td></td></tr>
<tr><td>Q. 06</td><td>a</td><td>Define heap. Explain the properties of heap along with its representation.</td><td>L2</td><td></td><td>10</td></tr>
<tr><td></td><td>b</td><td>Design Horspools algorithm for string matching. Apply Horspools algorithm to find the pattern BARBER in the text: JIM_SAW_ME_IN_A_BARBERSHOP</td><td>L3</td><td></td><td>10</td></tr>
<tr><td colspan="3">Module-4</td><td></td><td></td><td></td></tr>
<tr><td>Q. 07</td><td>a</td><td>Construct minimum cost spanning tree using Kruskals algorithm for the following graph.

</td><td>L3</td><td></td><td>10</td></tr>
<tr><td></td><td>b</td><td>What are Huffman Trees? Construct the Huffman tree for the following data.
<table><tr><td>Character</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>-</td></tr><tr><td>Probability</td><td>0.5</td><td>0.35</td><td>0.5</td><td>0.1</td><td>0.4</td><td>0.2</td></tr></table>Encode DAD-CBE using Huffman Encoding.</td><td>L3</td><td></td><td>10</td></tr>
</table>

OR

Q. 08 a Apply Dijkstra's algorithm to find single source shortest path for the given graph by considering S as the source vertex. — L3 — 10



b Define transitive closure of a graph. Apply Warshalls algorithm to compute transitive closure of a directed graph — L3 — 10

$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Module-5

Q. 09 a Explain the following with examples — L2 — 10

- i) P problem
- ii) NP Problem
- iii) NP- Complete problem
- iv) NP – Hard Problems

b What is backtracking? Apply backtracking to solve the below instance of sum of subset problem S={5,10,12,13,15,18} d=30 — L3 — 10

OR

Q. 10 a Illustrate N queen's problem using backtracking to solve 4-Queens problem — L2 — 10

b Using Branch and Bound technique solve the below instance of knapsack problem. — L3 — 10

Item	Weight	Value
1	2	12
2	1	10
3	3	20
4	2	5

Capacity 5