Modified

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**

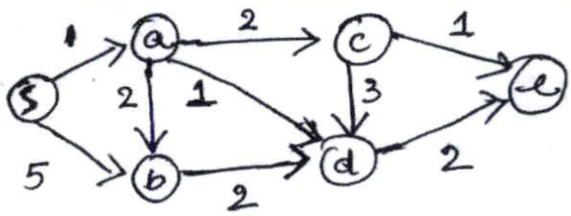


Fig.Q7(a)

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

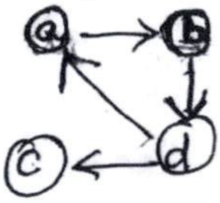


Fig.Q7(b)

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**

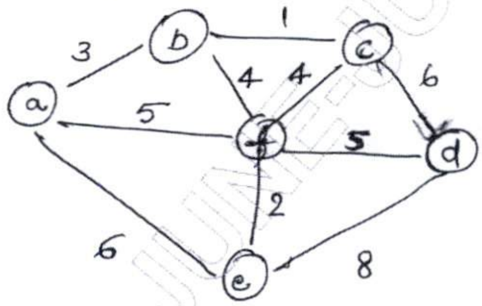


Fig.Q8(a)

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**

* * * * *

BCS401

Re: Madam, modifications of scheme and solutions

"Nirmala CR" <nirmala.cr@gmail.com> August 23, 2024 8:09 AM

To: boe@vtu.ac.in

Dear Sir,

I have checked the scheme and found appropriat. Kindly proceed with further process

Thanks and regards

NIRMALA C R

On Thu, 22 Aug 2024 at 12:57 PM, <boe@vtu.ac.in> wrote:

" APPROVED "

Registrar (Evaluation)
Visvesvaraya Technological University
BELAGAVI - 590018

2406BCS40170847

Visvesvaraya Technological University

Belagavi, Karnataka - 590 018.

Scheme & Solutions Signature of Scrutinizer

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

Question Number	Solution	Marks Allocated
1.a.	Algorithm Definition →	1M

Diagram :

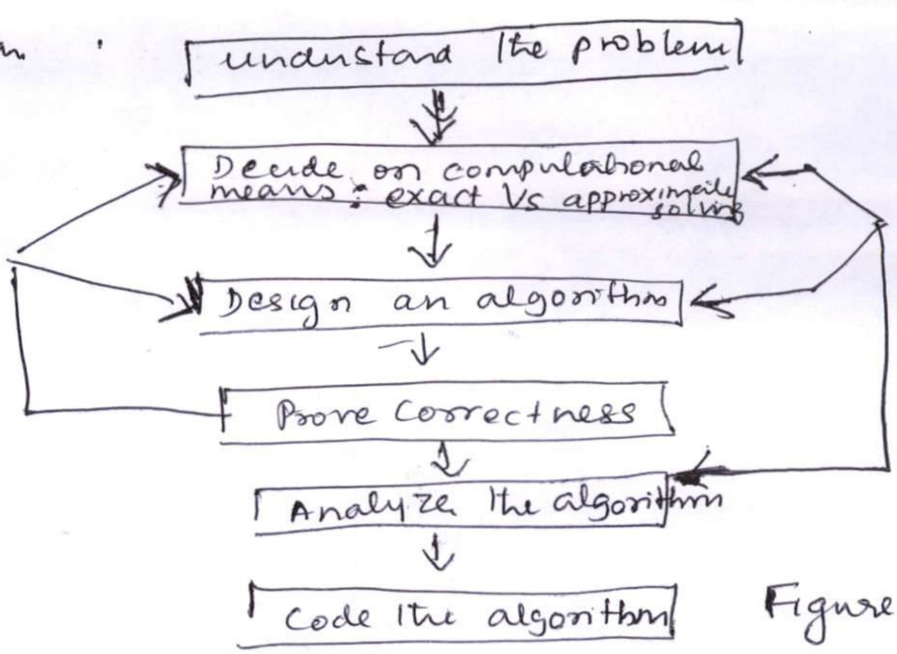


Figure → 3M

explanation of each step → 6M

total → 10M

Question Number	Solution	Marks Allocated
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~~

$$C_{avg}(n) = \left[1 \cdot \frac{p}{n} + 2 \cdot \frac{p}{n} + \dots + n \cdot \frac{p}{n}\right] + n(1-p)$$

$$= \frac{p}{n}\left[1 + 2 + \dots + n\right] + n(1-p)$$

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

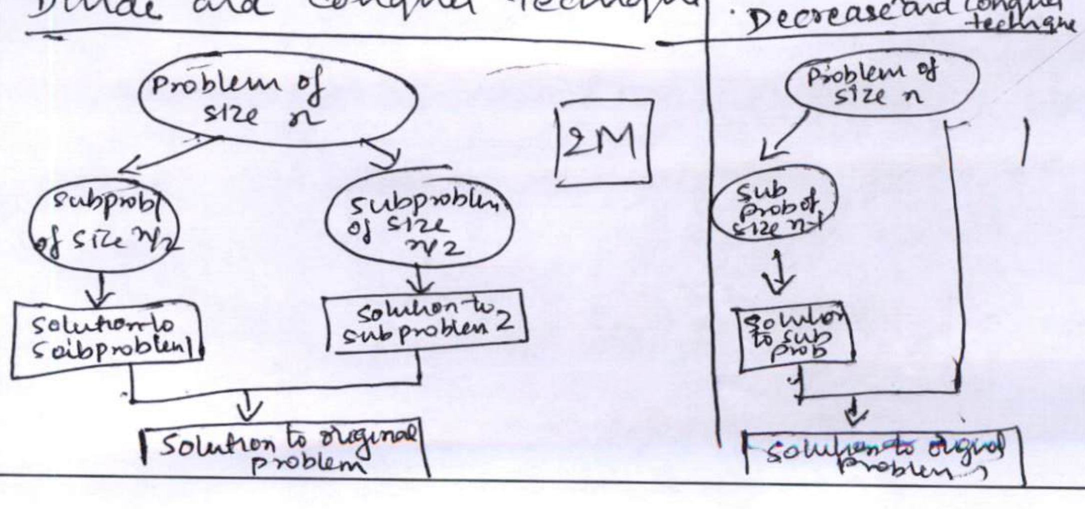
Question Number	Solution	Marks Allocated
	$= \frac{p}{n} \frac{n(n+1)}{2} + n(1-p) = \frac{p(n+1)}{2} + n(1-p)$	1M
	for successful search if $P = 1$ ~~Cave $\frac{(n+1)}{2}$~~ $C_{avg}(n) = \frac{n+1}{2} \rightarrow$	1M
	for unsuccessful search if $P = 0$ $\longrightarrow$ $C_{avg}(n) = \underline{\underline{n}}$	1M
	total	10M
2.a.	Asymptotic notations: Big-oh $(O) \rightarrow$ definition, figure & examples $\rightarrow$	3M
	Big-omega (Ω) - definition, figure & examples $\rightarrow$	3M
	Big-Theta (Θ) - definition, figure & example	4M
	total $\longrightarrow$	10M
2.b.	General plans listing: $\longrightarrow$ 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

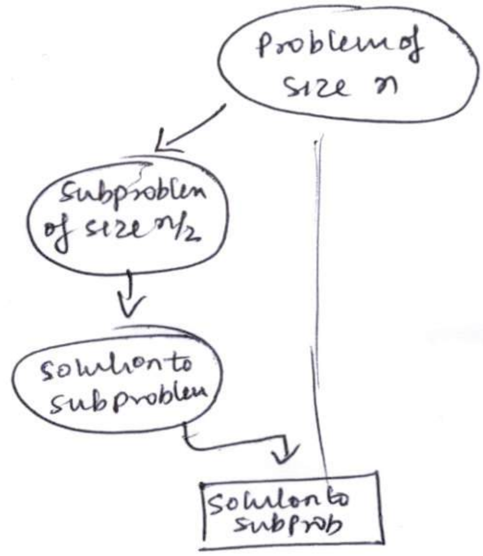
Subject Title : Analysis & Design of Algorithms **Subject Code :** BCS401

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



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
        $A[j+1] \leftarrow A[j]$
        $j \leftarrow j-1$
    $A[j+1] \leftarrow v$

$$C_{worst}(n) = \Theta(n^2)$$

$$C_{best}(n) = \Theta(n)$$

$$C_{avg}(n) = \frac{n^2}{4} \in \Theta(n^2)$$

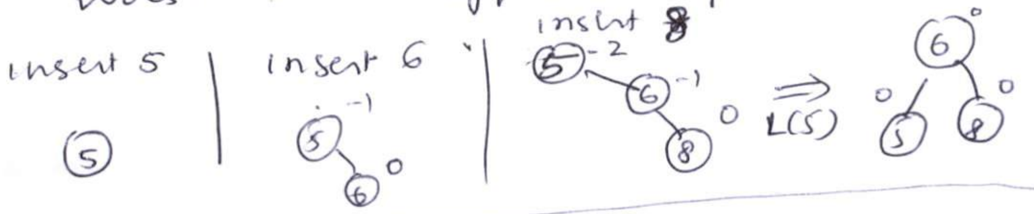
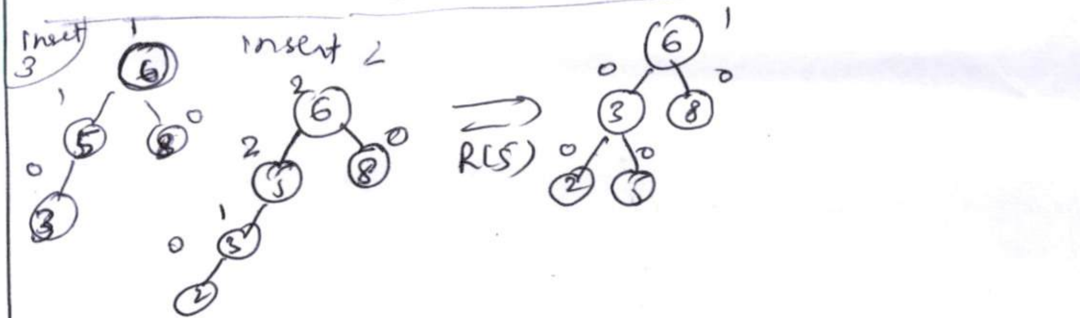
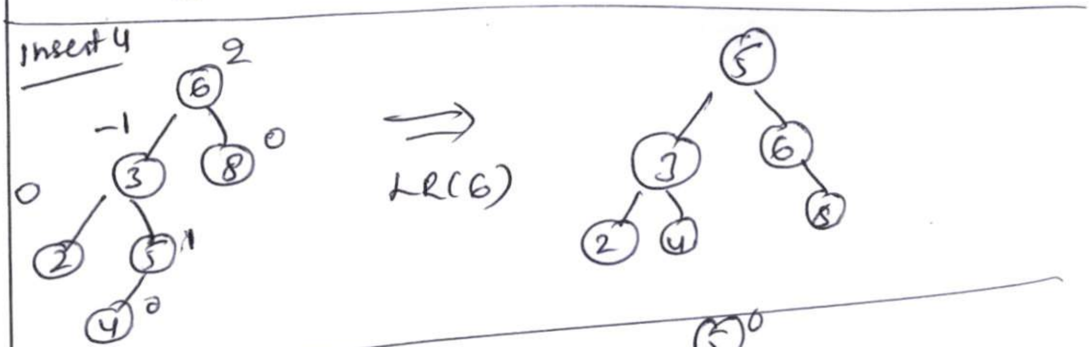
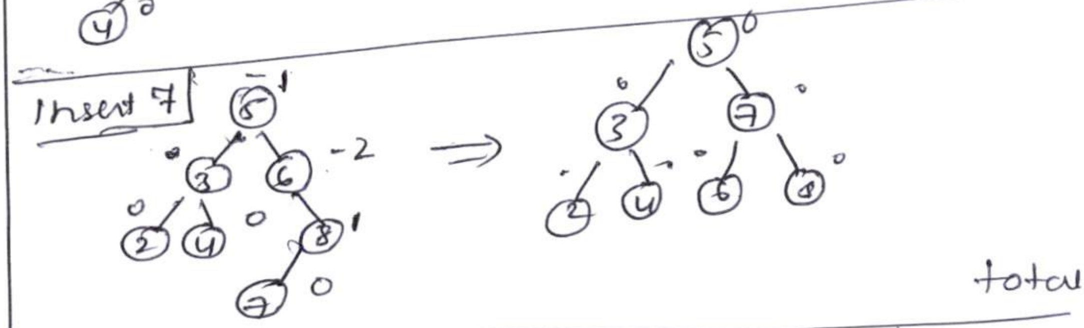
3M

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 → $\Theta(\log n)$ →	1M
		1M
		2M
		2M
		2M
	total	10
5.b.	Heap definition ~~& 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 ← ⌊n/2⌋ downto 1 do
    k ← i   v ← H[k]
    heap ← false
while not heap and 2*k ≤ n
    j ← 2*k
    if j < n
       if H[j] < H[j+1]  j ← j+1
    if v ≥ H[j]
       heap ← true
    else H[k] ← H[j] ; k ← j
H[k] ← v
```

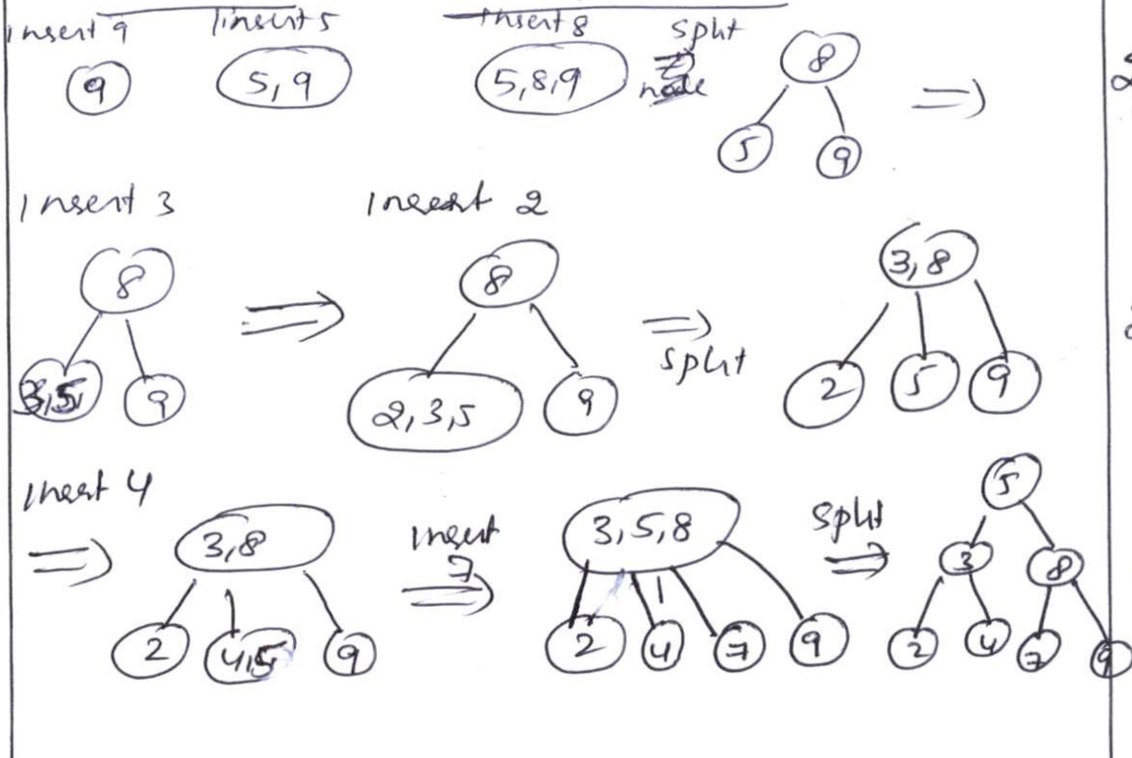
10

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

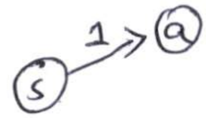
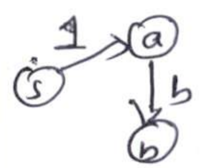
A	B	C	D	E	F	...	R	...	Z	_
4	2	6	6	1	6	6 ...	3	6 ..	6	6

<u>actual search</u> ~~4M~~ 8 / 4M

```
T: JIM_SAW_ME_IN_A_BARBERSHOP
P: BARBER
      BARBER
        BARBER
              BARBER
                 BARBER
                     BARBER
```

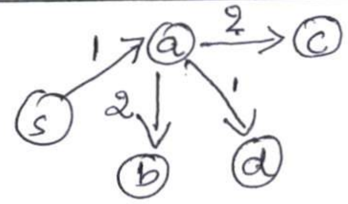
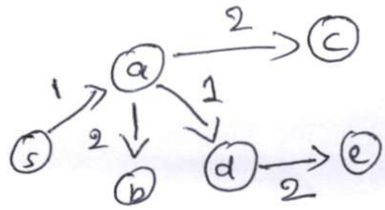
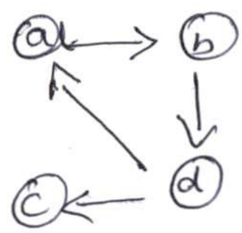
matched.

7. a.

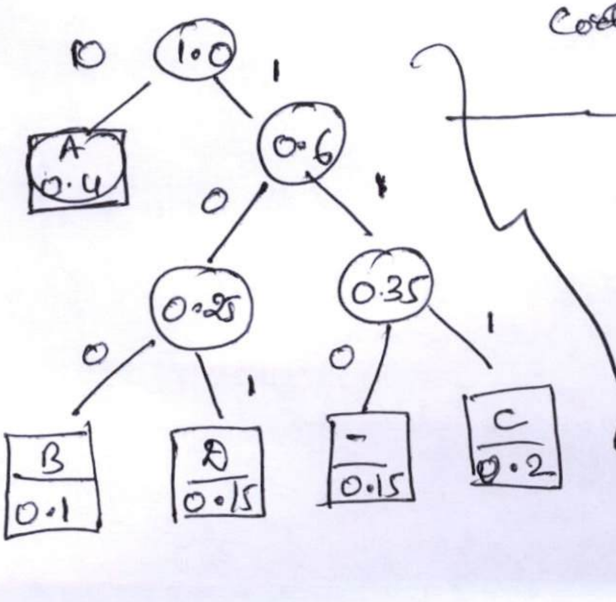
Tree vertices	Remaining Vertices	Illustrate	Marks
s(–,0)	<u>a(s,1)</u> b(s,5) c(–,∞) d(–,∞) e(–,∞)		1M ~~4M~~
a(s,1)	<u>b(a,3)</u> c(a,3) d(a,2) e(–,∞)		2M
~~b(a,3)~~	(b or c) any ~~thing~~ one can be selected to proceed as both are have same distance		
b(a,3)	c(a,3) <u>d(a,2)</u> e(–,∞)		2M

Cont...next page.

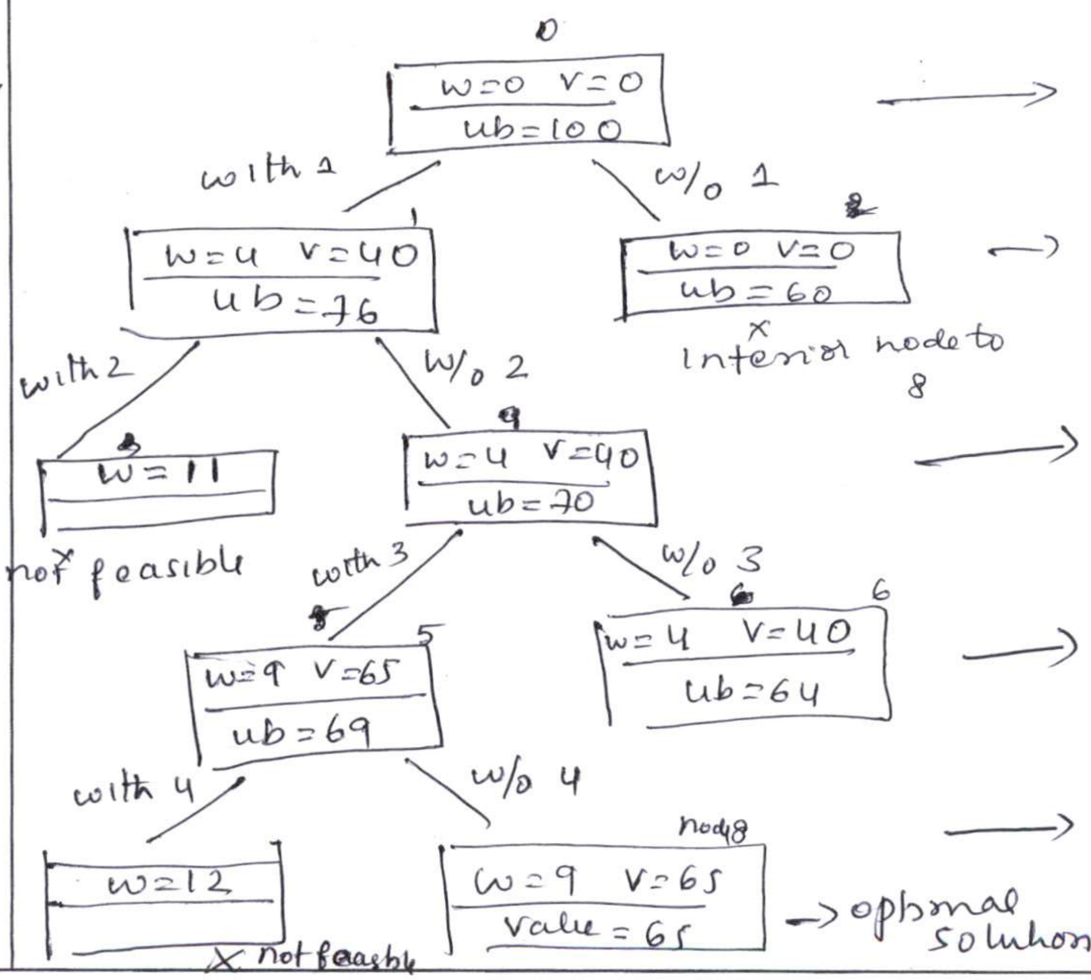
Subject Title : Analysis & Design of Algorithms Subject Code : BCS401

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) Shortest path s → a is 1 ~~s→a~~ s→a-b is 3 s→a→c is 3 | s→a-d is 2 s→a→d→e is 4	1M
7.b.	Definition of transitive closure	1M
	~~transit~~ Warshall's algorithm	3M
	Adjacency matrix  → $\begin{bmatrix} 0&1&0&0 \\ 0&0&0&1 \\ 0&0&0&0 \\ 1&0&1&0 \end{bmatrix}$	
	$R^{(0)} = \begin{bmatrix} 0&1&0&0 \\ 0&0&0&1 \\ 0&0&0&0 \\ 1&0&1&0 \end{bmatrix}$ $R^{(1)} = \begin{bmatrix} 0&1&0&0 \\ 0&0&0&1 \\ 0&0&0&0 \\ 1&1&1&0 \end{bmatrix}$	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 : BCS401

Question Number	Solution	Marks Allocated
	~~Avg # of bits per symbol = 2.2 bits~~	~~1M~~
	~~Fixed length encoding require = 3 bits~~	~~1M~~
	~~Compression ratio = $100\left(\frac{3-2.2}{3}\right)\times 1$~~ ~~$= \left(\frac{3-2.2}{3}\right)\times 100\,\%$~~ ~~$= 26.67\,\%$~~	~~2M~~
	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 x 1M	2M
	State space tree for subset sum problem →	8M
10.b	discrete knapsack problem Algorithm	4M
	Explanation	6M

" APPROVED "

Registrar (Evaluation)
Visvesvaraya Technological University
BELAGAVI - 590018