CBCS Scheme

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

15CS43

Modified

Fourth Semester B.E. Degree Examination, June/July 2017
Design and Analysis of Algorithms

Time: 3 hrs. Max. Marks: 80

Note: Answer FIVE full questions, choosing one full question from each module.

Module-1

1 a. Define algorithm. Explain asymptotic notations, Big O, big Omega, big theta notations. **(08 Marks)**

b. Explain general plan of mathematical analysis of nonrecursive algorithms with example. **(08 Marks)**

OR

2 a. Define time and space complexity. Explain important problem types. **(08 Marks)**

b. Illustrate mathematical analysis of recursive algorithm for towers of hanoii. **(08 Marks)**

Module-2

3 a. Explain concept of divide and conquer. Write merge sort algorithm. **(08 Marks)**

b. Write a recursive algorithm for binary search and also bring out its efficiency. **(08 Marks)**

OR

4 a. Illustrate the tracing of quick sort algorithm for the following set of numbers:
25, 10, 72, 18, 40, 11, 64, 58, 32, 9 **(08 Marks)**

b. List out the advantages and disadvantages of divide and conquer method and illustrate the topological sorting for the following graph.

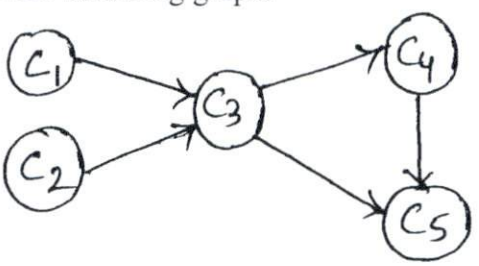


Fig.Q4(b) **(08 Marks)**

Module-3

5 a. Explain Greedy criterion. Write a Prim's algorithm to find minimum cost spanning tree. **(08 Marks)**

b. Sort the given list of numbers using heap sort: 2, 9, 7, 6, 5, 8. **(08 Marks)**

OR

6 a. Write an algorithm to find single source shortest path. **(08 Marks)**

b. Construct a Huffman tree and resulting code word for the following:

Character	A	B	C	D	-
Probability	0.35	0.1	0.2	0.2	0.15

Encode the words DAD and ADD. **(08 Marks)**

Module-4

7 a. Explain the concept of dynamic programming, with example. **(08 Marks)**

b. Trace the following graph using Warshall's algorithm.

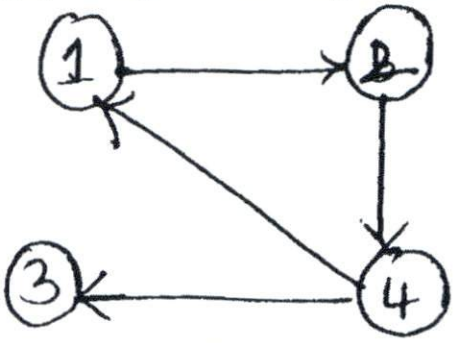


Fig.Q7(b) **(08 Marks)**

OR

8 a. Explain Multistage graphs with example. Write multistage graph algorithm to forward approach. **(08 Marks)**

b. Solve the following instance of Knapsack problem using dynamic programming. Knapsack capacity is 5.

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

(08 Marks)

Module-5

9 a. Explain backtracking concept. Illustrate N queens problem using backtracking to solve 4-Queens problem. **(08 Marks)**

b. Solve subset sum problem for the following example, s = {3, 5, 6, 7} and d = 15. Construct a state space tree. **(08 Marks)**

OR

10 a. Explain the concept of branch and bound and solve assignment problem for the following and obtain optimal solution.

		Job1	Job2	Job3	Job4
Person	a	9	2	7	8
	b	6	4	3	7
	c	5	8	1	8
	d	7	6	9	4

(08 Marks)

b. Explain LC Branch and Bound and FIFO branch and bound. **(08 Marks)**

* * * * *

Visvesvaraya Technological University
Belagavi, Karnataka – 590 018.

Scheme & Solutions

Subject Title : DAA (Design and Analysis of Algorithm Subject Code : 15CS43

Signature of Scrutinizer

Question Number	Solution	Marks Allocated
1a	Define Algorithm — MODULE 1 An Algorithm is a sequence of unambiguous instructions for solving a problem i.e for obtaining a required o/p for any legitimate i/p in a finite amount of time	1M
	Asymptotic Notation Study of functions of a parameter n as n becomes larger & larger without bound. To analyse algorithms running time by identifying its behavior as i/p size for algo increases	1M
→	O Bigoh $f(n) = O(g(n)), f(n) \leq c * g(n)$	3×2 = 6M.
→	Ω Big Omega $f(n) = \Omega(g(n)), f(n) \geq c \cdot g(n)$	
→	Θ Big Theta $f(n) = \Theta(g(n)), c_1(g(n) \leq f(n) \leq c_2(g(n))$	(1+1+6=8M)
1b	General Plan Decide on a parameter indicating i/p size Identify algo's basic operation Check no of times basic operation Executed Additional property of worst, best, avg cases investigated	4M
	Using std formulas & rules establish order of growth Any one non recursive algo with efficiency calculation	4M
		(4+4=8M)
	MODULE 2 OR	
2a	Time Efficiency – Indicates how fast an algo in question runs Space Efficiency – Indicates how much space algo requires to run	2M (1×2)
	Important problem types. – Sorting – Searching – String processing – Graph problems – Combinatorial Problems – Geometric & Numerical Problems	6M (1×6)
		(2+6=8M)

Dr. M RAVISHANKAR
BOE Chairman CS/IS
Principal
Vidya Vikas Institute of Engineering & Technology
Mysuru-570028 Mobile 9845550525
Email:boecsismysuru@gmail.com

OR

"APPROVED"
Registrar (Evaluation)
Visvesvaraya Technological University
BELAGAVI-590018

Subject Title : DAA

Subject Code : 15CS43

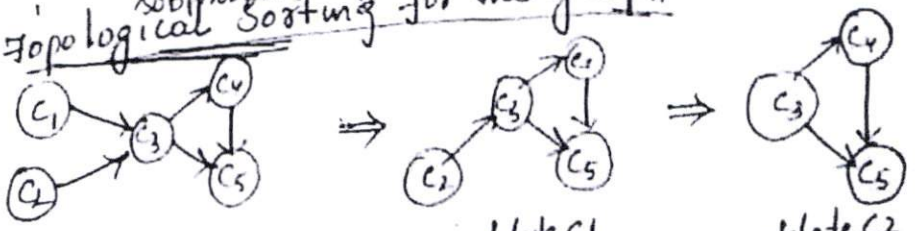
Question Number	Solution	Marks Allocated
2b	Towers of hanoi : Mathematical analysis $Mv(n) = 2Mv(n-1)+1$ $= 2[2Mv(n-2)+1]+1 = 2^2Mv(n-2)+2+1$ $= 2^2[2Mv(n-3)+1]+2+1 = 2^3Mv(n-3)+2^2+2+1$ $\approx 2^4Mv(n-4)+2^3+2^2+2+1$ $2^iMv(n-i)+2^{i-1}+2^{i-2}+\dots+2+1 \approx 2^iMv(n-i)+2^i-1$ Initial cond $n-i=1$, replace i by $n-1$ $2^{n-1}M(1)+2^{n-1}-1 = 2^{n-1}+2^{n-1}-1 \approx 2^n-1$	8

3a	Divide & conquer Method
	The divide & conquer strategy suggests splitting the inputs into k distinct subsets $1<k\leq n$, yielding k subproblems these subproblems must be solved & then a method must be found to combine subsolns into a soln of the whole.
	Algo Divide & conquer(P)
	{ if small(P) then return S(P); else divide P into small instances $P_1, P_2 \dots P_k$ $k\geq 1$; Apply Divide & conquer to each of these subproblems return combine(Divide & conquer(P1), Divide and conquer(P_2) ... Divide and conquer P_k)); }
	MergeSort Algo
	Algo MergeSort(low,high)
	{ if (low < high) then
	{mid = (low + high)/2
	MergeSort (low, mid)
	MergeSort (mid+1, high)
	Merge (low, mid, high)
	}
	}
	Algo Merge (low, mid, high)
	i ← low, j ← mid+1, k ← low
	while (i<=mid && j<=high)
	if (a[i] < a[j]) then
	c[k] ← a[i]
	i++; k++;
	- conti
	else
	c[k] ← a[j]
	j++ ; k++
	endif
	end while
	while (i<=mid)
	c[k] ← a[i]
	k++; i++
	end while
	while (j<=high)
	c[k] ← a[j]
	k++; j++
	End while
	-for i ← low to high
	a[i] ← c[i]
	end for
	Efficiency $n\log_2 n$
		2M
		6M

Dr. M RAVISHANKAR
BOE Chairman CS/IS
Principal
Vidya Vikas Institute of Engineering & Technology
Mysuru-570028 Mobile 9845550525
Email:boecsismysuru@gmail.com

Subject Title : DAA

Subject Code : 15CS43

Question Number	Solution	Marks Allocated
3b	Algo BSR(a,l,h,x) { if (h=l) then { if x = a[l]) then return l; else return 0 } else mid = (l+h)/2 if (x = a[mid]) then return mid; elseif (x < a[mid]) then return BSR(a,l,mid-1,x); else return BSR(a,mid+1,h,x) } Efficiency. $\Theta(\log(n))$ worst, $\Theta(\log(n))$ Avg, $\Theta(1)$ Best	6M 2M (6+2 = 8M)
4a	Quick Sort Trace. 25 10 72 18 40 11 64 58 32 9 25 10 9 18 40 11 64 58 32 72 25 10 9 18 11 40 64 58 32 72 11 10 9 18 25 40 64 58 32 72 11 10 9 18 → 40 64 58 32 72 11 10 9 18 → 40 32 58 64 72 9 10 11 18 → 32 40 58 64 72 9 10 → 58 64 72 9 10 → 64 72	8M.
4b	Advantages & Disadvantages of Divide & conquer Adv – Allows to solve difficult problems Ex Towers of Hanoi, i.e. reduces degree of difficulty. It uses memory cache efficiently Dis Adv – Sometimes it becomes more complicated than basic iterative method. Recursion is slow process. problem broken into subproblem & subproblem occurs many times Topological Sorting for the graph   Soln C1 C2 C3 C4 C5 (2+6=8M or C2 C1 C3 C4 C5	2M. 6M.

Dr. M RAVISHANKAR
BOS Chairman CS/IS
Principal
Vidya Vikas Institute of Engineering & Technology
Mysuru-570028 Mobile 9845550525
Email:beecsismysuru@gmail.com

5a → Greedy method

we attempt to construct an optimal solution in stages. At each stage we make a decision that appears to be the best at the time. A decision made in one stage is not changed in a latter stage, so each decision should assure feasibility. The criterion used to make greedy decision at each stage is called Greedy criterion. 3M

```
Algo Greedy(a,n)
{ Solution = ∅ ; for i=1 to n do { for x = Select(a);
  if feasible(Solution, x) then Solution = Union(Solution, x)}
  return Solution ;
}
```

→ Prims Algorithm.

// construct min spanning tree G(V,E)

$V_T \leftarrow \{v_0\}$; $E_T \leftarrow \emptyset$; for $i \leftarrow 1$ to $|V|-1$ do

find a min weight edge $e^* = (v^*, u^*)$ among all the edges (v,u) such that v is in V_T & u is in $V-V_T$. 5M.

$V_T \leftarrow V_T \cup \{u^*\}$

$E_T \leftarrow E_T \cup \{e^*\}$.

return E_T.

5b. Sort the list 2, 9, 7, 6, 5, 8 using heap sort.

Stage 1 (heap constn)

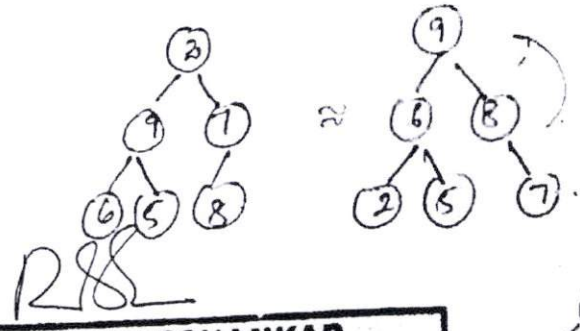
2 9 7 6 5 8
2 9 8 6 5 7
2 9 8 6 5 7
9 2 8 6 5 7
9 6 8 2 5 7.

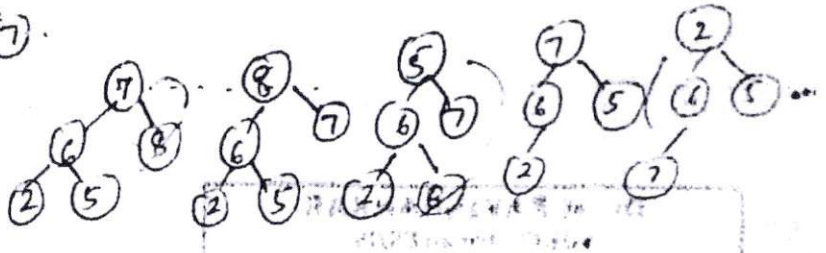
Stage 2 Max deletion

9 6 8 2 5 7
7 6 8 2 5 | 9
8 6 7 2 5
5 6 7 2 | 8
7 6 5 2
2 6 5 | 7
6 2 5
5 2 | 6
2 5
2 | 5
| 2

| 2 | 5 | 6 | 7 | 8 | 9 |

8M.





Subject Title : DAA

Subject Code : 15CS43

Question Number	Solution	Marks Allocated

6a	Single Source Shortest Path (Dijkstras (G, s)
	// V_T : Source vertex & fringe contains vertices adjacent
	// I/P G = (V, E)
	// O/P The length d_v of shortest path from s to v & the penultimate vertex P_v for every vertex v in V.
	// Initialize Q
	for every vertex v in V do ; $d_v \leftarrow \infty$; $P_v \leftarrow$ null
	insert (Q, v, d_v) ; $d_s \leftarrow 0$; Decrease(Q, s, d_s)
	$V_T \leftarrow \emptyset$
	for i ← 0 to |V| − 1 do
	$u^* \leftarrow$ Delete min (Q)
	$V_T \leftarrow V_T \cup \{u^*\}$
	for every vertex u in $V - V_T$ that is adjacent to u^* do
	if $d_{u^*} + w(u^*, u) < d_u$
	$d_u \leftarrow d_{u^*} + w(u^*, u)$; $P_u \leftarrow P_{u^*}$
	Decrease (Q, u, d_u).
		8M

6b	Huffman Tree Construction character | A B C D – Probability | 0.35 0.1 0.2 0.2 0.15 Code word | 11 100 00 01 101 encode DAD 01 11 01 ADD 11 01 01	8M

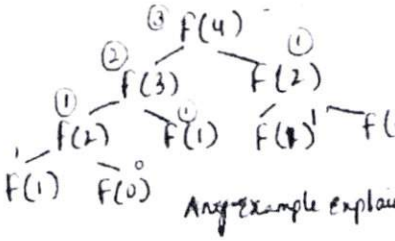
Dr. M RAVISHANKAR
BOE Chairman CS/IS
Principal
Vidya Vikas Institute of Engineering & Technology
Mysuru-570028 Mobile 9845558525
Email:boecsismysuru@gmail.com

7a Dynamic Programming Concept

Dynamic programming is a technique for solving problems with overlapping subproblems. These subproblems arise from a recurrence relating a solution to a given problem with solutions to its smaller subproblems of the same type. 4M

Ex fibonacci 0,1,1,2,3,5,8,13,21,34

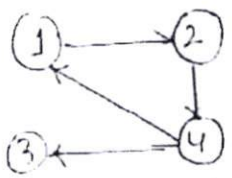
$f(n) = f(n-1) + f(n-2)$ $f(0)=0$ $f(1)=1$ 4M.



(4+4) = 8M.

Any example explaining concept.

7b) Warshall's Algo Trace



$R^{(0)}$

	1	2	3	4
1	0	1	0	0
2	0	0	0	1
3	0	0	0	0
4	1	0	1	0

$R^{(1)}$

	1	2	3	4
1	0	1	0	0
2	0	0	0	1
3	0	0	0	0
4	1	1	1	0

$R^{(2)}$

	1	2	3	4
1	0	1	0	1
2	0	0	0	1
3	0	0	0	0
4	1	1	1	1

$R^{(3)}$

	1	2	3	4
1	0	1	0	1
2	0	0	0	1
3	0	0	0	0
4	1	1	1	1

$R^{(4)}$

	1	2	3	4
1	1	1	1	1
2	1	1	1	1
3	0	0	0	0
4	1	1	1	1

8M

8a Multistage graphs with example.

→ The multistage graph problem is to find a min cost path from s to t where s is source & t is sink. Each set V_i defines a stage in the graph. Because of constraint on it, every path from s to t starts in stage 1, goes to stage 2 then to stage 3 & 4m then to stage 4 & so on & terminates in stage k.

→ Algo FGraph(G,k,n,p) 4m

```
{ cost[n] := 0.0;
  for j = n-1 to 1 step -1 do
  // compute cost[j]
     Let r be a vertex such that <j,r> is an edge of G and c[j,r] + cost[r] is min
     cost[j] = c[j,r] + cost[r];
     d[j] = r;
  // find min cost s-path
  P[1] = 1; P[k] = n; for j = 2 to k-1 do
     P[j] = d[P[j-1]] }
```

(4+4m) = 8M

8b Knapsack capacity = 5

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

Dr. M RAVISHANKAR
HOE Chairman CS/IS
Principal
Vidya Vikas Institute of Engineering & Technology
Mysuru-570028 Mobile 9845550525
Email:hoecsisamysuru@gmail.com

Subject Title : DAA

Subject Code : 15CS43

Question Number	Solution	Marks Allocated

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0	10	15	25	30	(37)

$W_1 = 2 \quad V_1 = 12$

$W_2 = 1 \quad V_2 = 10$

$W_3 = 3 \quad V_3 = 20$

$W_4 = 2 \quad V_4 = 15$

8 M

optimal soln – 37.

Trace $V[i,j] = \max V[i-1,j], V_i + V[i-1, j-w_i]$ if $j-w_i \geq 0$

$V[i-1,j]$ if $j-w_i < 0$

$V(1,1) = 0$

$V(1,2) = V(0,2), 12 + V(0,0)$

$= 12$

$V(1,3) = V(0,3), 12 + V(0,1)$

$= 12$

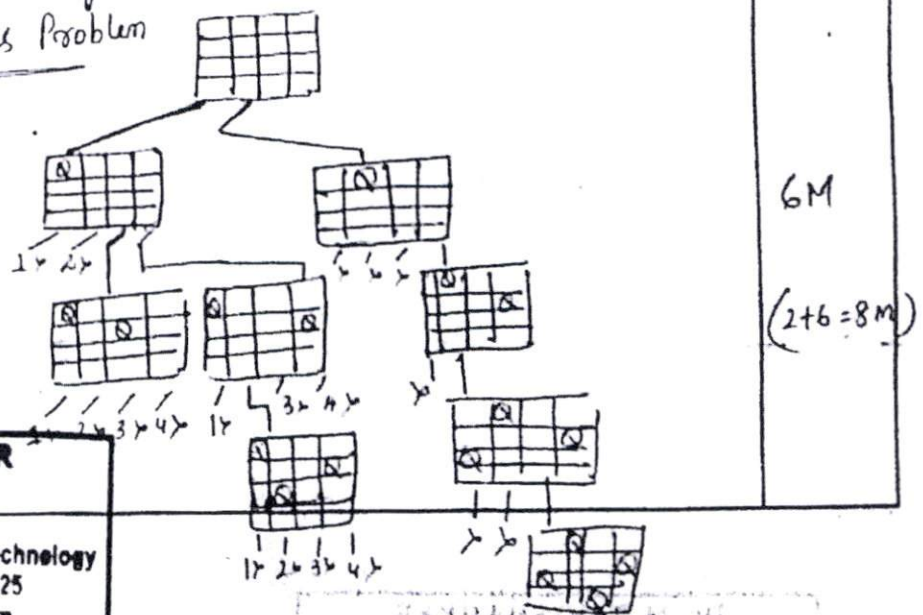
9a

Backtracking Concept

The principal idea is to construct solution one component at a time, it continues constructing a solution at the problems constraint (rule) is satisfied. If it violates the rule, the algo backtracks to replace with next option. It is implemented by constructing a tree called State Space tree

2M

4 Queens Problem



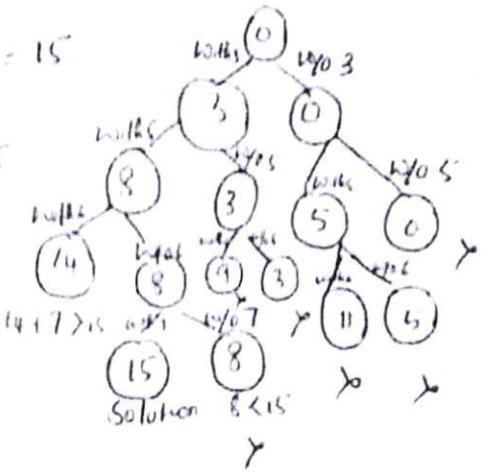
6M

(2+6 = 8M)

Dr. M RAJESHANKAR
BOE Chairman CS/IS
Vidya Vikas Institute of Engineering & Technology
Mysuru-570028 Mobile: 9845550525
Email:boecsismysuru@gmail.com

9b $S = \{3, 5, 6, 7\}$ $d = 15$

Solution $\{3, 5, 7\} = 15$



8M

10a Branch & Bound concept

It is an improvement over exhaustive search. Here, they construct candidate solutions, if no potential values of the remaining components can lead to a solution, the remaining components are not generated at all. 3M.

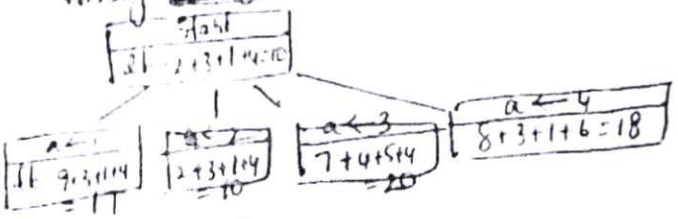
Branch & Bound searches the tree in either Breadth first search or least cost manner

Branch: Way in which we search the state space tree

Bound: means assigning bound functions to each node.

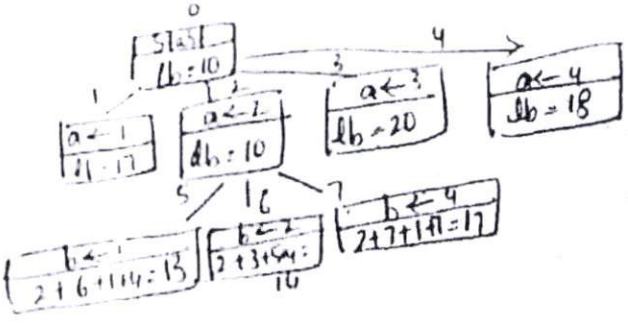
Assignment problem

→



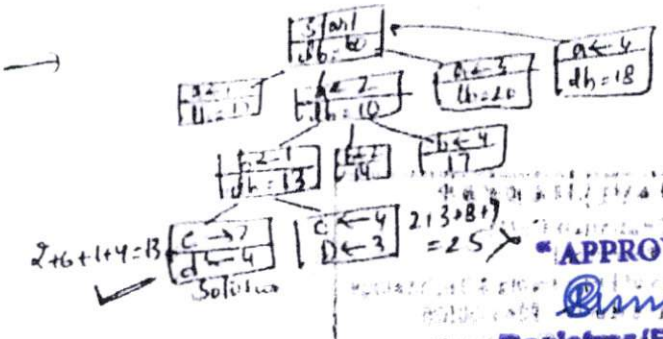
5M

→



(3+5=8M)

→



Dr. M RAVISHANKAR
BOE Chairman CS/IS
Principal
Vidya Vikas Institute of Engineering & Technolog
Mysuru-570028 Mobile 9845558525
Email:boecsismysuru@gmail.com

10b LC Branch & Bound (4 M)

FIFO Branch & Bound (4 M)

(4+4=8M)

"APPROVED"
Registrar (Evaluation)
Visvesvaraya Technological University
BELAGAVI - 590018