23ICPC207

MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

(A Unit of Rajalaxmi Education Trust®, Mangalore)
Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE, New Delhi
Accredited by NAAC with A+ Grade & ISO 9001:2015 Certified Institution

Model Question Paper

Fourth Semester BE Degree Examination

Design and Analysis of Algorithms

Time: 3 Hours **Max. Marks: 100**

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

		Module -1	**M**	**L**	**C**
Q1	a.	Explain the need for algorithm efficiency. Compute the Best-case, Worst-case, and Average-case time complexity of sequentially searching the array of numbers {20, 10, 15, 5, 25}	09	L3	CO1

	b.	Compute the time complexity of the code segment given below:
		main ( )
		{
		    for(i=1; i<=n; i++)
		    {
		        for(j=1; j<=n; j=2*j)
		        {
		            x = y+z;
		        }
		    }
		}
			04	L3	CO1

	c.	Give the definition & mathematical representation of Big Oh (O) notation and test if f(n) = O(g(n)) when f(n) = n+20 and g(n) = n. Compute the Omega (Ω) notation of $f(n) = 7n^3 + 5n^2 + 2n + 4$.	07	L3	CO1
		OR			

Q2	a.	Draw a flowchart depicting the fundamental stages of problem solving and briefly explain the stages.
		Compute the time complexity of the code segment given below
		main ( )
		{
		    k=2;
		    while (k<=n)
		    {
		        k = k*k;
		    }
		}
			09	L3	CO1

	b.	Consider three algorithms X, Y and Z. Their respective run-time complexity is given as follows: $T_X = 5n$, $T_Y = 5\log_{10} n$, and $T_Z = n^2$. Compute the rates of growth of each algorithm when the input is scaled from n=10 to n=1000 and state which of the three algorithms is better looking at their rates of growth.	04	L3	CO1
	c.	Give the definition and mathematical representation of Big Omega (Ω) notation. Let $f(n) = n$ and $g(n) = n^2$. Test if $f(n) = \Omega(g(n))$.	07	L3	CO1

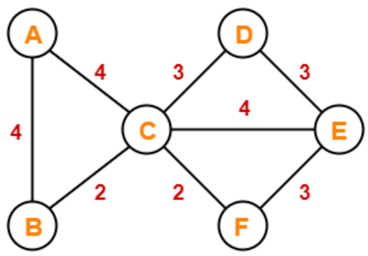
		Consider that $t(n) = (2x^3 + 13 \log_2 x) / 7n^2$ for an algorithm A. Prove that t(n) of algorithm A is O(n).			
		Module- 2			

Q3	a.	Develop the recurrence relation for the following function
		
		`void DAA(int n)`
		`{`
		`    if(n>0)`
		`    {`
		`        printf("%d", n);`
		`        DAA(n-1);`
		`        DAA(n-1);`
		`    }`
		`}`
		
		Solve the recurrence relation for the above function using backward substitution method.
			05	L3	CO2

	b.	Consider the following recurrence $T(n) = \begin{cases} 1, & n = 0 \\ T(n-1) + n, & n > 0 \end{cases}$ Using the recurrence-tree method, determine the final asymptotic complexity of the tree. Compute time complexity of Bubble sort algorithm. Use Bubble sort algorithm to sort the array of numbers {55, 16, 33, 44, 25}	08	L3	CO2
	c.	In the algorithm of sorting by Merge sort, write the formal algorithm for the merge step. Use Merge sort to sort the array of numbers {15, 6, 5, 17, 22, 8, 12, 21}. Write the recurrence equation of Merge sort and use it to find the time complexity of Merge sort by Master Theorem.	07	L3	CO2
		OR			

Q4	a.	Develop the recurrence relation for the following function
		
		`void DAA (int n)`
		`{`
		`    if(n>1)`
		`    {`
		`        for (i=0; i<n; i++)`
		`        {`
		`            printf("%d", n);`
		`        }`
		`        DAA(n/2);`
		`        DAA(n/2);`
		`    }`
		`}`
		
		Solve the recurrence relation for the above function using backward substitution method.
			05	L3	CO2

	b.	Solve the following recurrence relations using master theorem a) $T(n)=2T(n/2)+1$ b) $T(n)=4T(n/2)+n^2$ c) $T(n)=2T(n/2)+n^2 \log n$ d) $T(n)=2T(n/2)+n/\log n$	08	L3	CO2

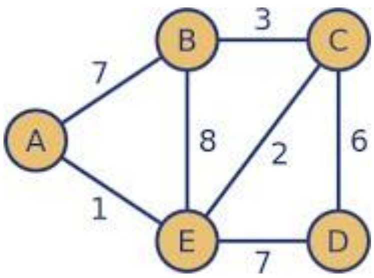
<table>
<tr><td></td><td></td><td>Compute time complexity of Selection Sort algorithm. Use Selection sort algorithm to sort the array of numbers {72, 23, 33, 36, 21}</td><td></td><td></td><td></td></tr>
<tr><td></td><td>c.</td><td>In the algorithm of sorting by Quick sort, write the formal algorithm for Hoare Partitioning and use it to sort the array of numbers {7, 12, 5, 6, 27}. Write the recurrence equation of Quick sort and use it to find the Best-case time complexity of Quick sort by Master Theorem.</td><td>07</td><td>L3</td><td>CO2</td></tr>
<tr><td colspan="6">Module - 3</td></tr>
<tr><td rowspan="3">Q5</td><td>a.</td><td>Write the formal algorithm for the Insertion sort and compute its Best-case, Worst-case, and Average-case time complexity. Use Insertion sort algorithm to sort the array of numbers {75, 35, 45, 10, 20, 50}.</td><td>09</td><td>L3</td><td>CO3</td></tr>
<tr><td>b.</td><td>Consider the jobs and profit shown in the table. Find the optimal set of jobs that can be scheduled so that the profit is maximized.
<table>
<tr><th>Job</th><th>Deadline</th><th>Profit</th></tr>
<tr><td>1</td><td>2</td><td>70</td></tr>
<tr><td>2</td><td>1</td><td>12</td></tr>
<tr><td>3</td><td>2</td><td>18</td></tr>
<tr><td>4</td><td>1</td><td>35</td></tr>
</table>
Construct the Huffman tree for the following data and obtain Huffman Code for all symbols.
<table>
<tr><td>Symbol</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td></tr>
<tr><td>Frequency</td><td>21</td><td>11</td><td>12</td><td>17</td><td>39</td></tr>
</table></td><td>07</td><td>L3</td><td>CO3</td></tr>
<tr><td>c.</td><td>Make use of Prim's algorithm to find the minimum cost spanning tree for the given graph starting from vertex ‘A’.

</td><td>04</td><td>L3</td><td>CO3</td></tr>
<tr><td colspan="6">OR</td></tr>
<tr><td>Q6</td><td>a.</td><td>Write the formal algorithm for the Binary Search and compute its Best-case, Worst-case, and Average-case time complexity. Use Binary Search algorithm to search the numbers 30 and 13 in the array of numbers {4, 12, 16, 25, 27, 30, 33}.</td><td>09</td><td>L3</td><td>CO3</td></tr>
</table>

b. Let there be a knapsack with capacity W=35. Let there be three items whose profit and weight are given in the table. Find the optimal order for loading the items in the given knapsack. — 07 | L3 | CO3

Items	Weight	Profit
1	18	54
2	19	38
3	15	60

Assume that there are 3 programs L1, L2 & L3 whose lengths are 5, 10 and 3 respectively. Find the optimal order of storing these programs on a tape using brute force approach.

c. Make use of Dijkstra's algorithm to find the shortest path from ***source vertex 'A'*** to all other vertices. — 04 | L3 | CO3

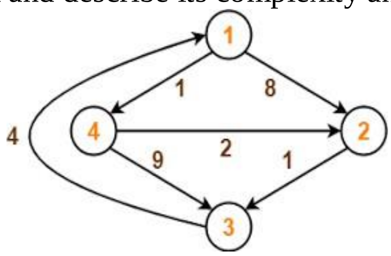


Module - 4

Q7

a. Write the formal algorithm of heap sort and describe its complexity analysis. Sort the array of numbers {7, 5, 2, 8, 1, 3, 4} in ascending order using Heap Sort by constructing an appropriate Heap tree for this purpose. — 07 | L3 | CO4

b. Find the shortest path between all pairs of vertices for the given graph using the Floyd-Warshall algorithm and describe its complexity analysis. — 07 | L3 | CO4

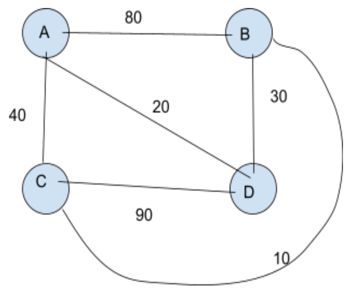


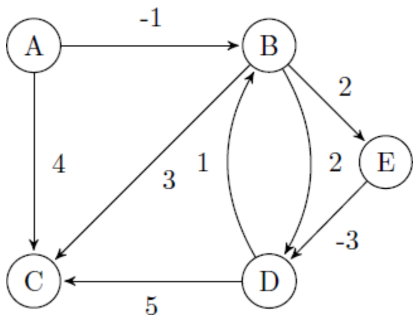
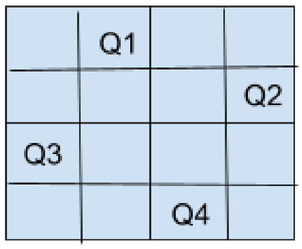
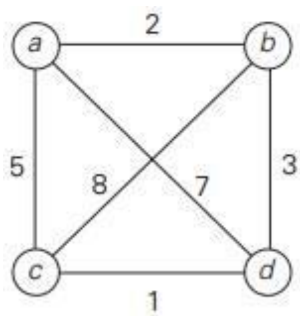
c. Find all sum of subsets for n=4, (w1, w2, w3, w4) = (11, 13, 24, 7) and M=31. Construct the state space tree of the solution. — 06 | L3 | CO4

OR

Q8

a. Solve the Travelling Salesperson Problem (TSP) using Dynamic Programming for the given graph with source as A and describe its complexity analysis. — 07 | L3 | CO4



	b.	Compute the single source shortest path for the graph from source vertex A using the Bellman Ford Algorithm and describe its complexity analysis. 	07	L3	CO4
	c.	Make use of this solution of the 4-Queen's problem to construct the state space tree leading up to this solution. 	06	L3	CO4
		Module - 5			
Q9	a.	Write the following string matching algorithms and describe their complexity analysis. a) naïve string matching of both b) Rabin-Karp string matching Make use of the Rabin-Karp string matching algorithm to search the pattern "DBA" in the text "CCACCAAEDBA"	08	L3	CO5
	b.	Make use of the Brute force string matching algorithm to search the pattern "MET" in the text "COMET". State the differences between NP, NP-Complete and NP-Hard. Explain Satisfiability Problem	06	L3	CO5
	c.	Solve the Traveling Salesperson Problem (TSP) for the graph below using Branch and Bound. 	06	L3	CO4
		OR			
Q10	a.	Write the formal algorithm for Knuth-Morris-Pratt string matching and describe its complexity analysis. Make use of the Knuth-Morris-Pratt string matching algorithm to search the pattern "YYZY" in the text "YYZYYXYYWYYZY".	08	L3	CO5
	b.	Make use of the Brute force string matching algorithm to search the pattern "ICE" in the text "DEVICE". Show that the Clique Decision Problem is NP-Complete.	06	L3	CO5

	c.	Solve the following Knapsack Problem using Branch and Bound with knapsack capacity W=.6.	06	L3	CO4

Items	**Weight**	**Value**
1	2	8
2	3	6
3	2	4

-*****-