USN	0	1	J	C	E	2	1	C	S	1	2	1

"20CS420"

JSS MAHAVIDYAPEETHA

JSS SCIENCE AND TECHNOLOGY UNIVERSITY, MYSURU

Computer Science and Engineering

IV Semester : CIE-1(Event-I)

Design and Analysis of Algorithms

Duration:1 Hr. Date: 05.05.2023 Max. Marks:20

NOTE: PART-A is Compulsory, PART-B has internal choice

PART-A

Q.NO	CO	CD/PI	QUESTION	MARKS
1.	CO1	L2 1.7.1	a) List important types of problems and discuss with examples, the issues associated with the sorting problem. b)What are asymptotic notations? Discuss with examples different asymptotic notations.	03+07

PART-B

2.	CO1	L2 2.5.3	a) Design iterative algorithm to find the element uniqueness property in an unsorted array of 'n' elements and analyze the efficiency. b) Appraise the use of limit based approach to compare the order of growth of the functions and apply the same to compare the growth of the following functions i) logn and nlogn ii) 2^n and n!	05+05
OR				
3.	CO1	L2 2.5.3	a) Design an iterative algorithm to compute the sum of the squares of first 'n' natural numbers and establish the order of growth of the algorithm's basic operation count. b) Consider the following algorithm. *Algorithm Secret(A[0..n − 1])* *//Input: An array A[0..n − 1] of n real numbers* *T1 ← A[0]; T2 ← A[0]* *for i ← 1 to n − 1 do* *if A[i] < T1* *T1← A[i]* *if A[i] > T2* *T2 ← A[i]* *return T2 – T1* a. What does this algorithm compute? b. What is its basic operation c. What is the efficiency class of this algorithm? d. Is it possible to reduce the number of basic operations? Justify your answer.	05+05

USN	0	1	J	C	E	2	1	C	S	1	2	1

"20CS420"

JSS MAHAVIDYAPEETHA
JSS SCIENCE AND TECHNOLOGY UNIVERSITY, MYSURU

Computer Science and Engineering
IV Semester : CIE-3(Test-2)

Design and Analysis of Algorithms

Duration:1 Hr. Date: 13.06.2023 Max. Marks:20

NOTE: PART-A is Compulsory, PART-B has internal choice

PART-A

Q.NO	CO	CD/PI	QUESTION	MARKS
1.	CO2	L2 2.5.2	a) Discuss the strengths and weaknesses of brute force design strategy. b) With the tree of recursive calls, trace the Quick sort algorithm to sort the list : 'M','E','R','G','E','S','O','R','T' in alphabetical order. What is the best case input for the quick sort? Setup a recurrence relation for the same and solve it to find the order of growth of the basic operation.	3+7

PART-B

2.	CO2 CO3	L2 ,L3 2.5.3	a) Write algorithms to find a^n (a>0 and n is a positive integer) with divide and conquer and decrease by two strategies. Perform comparative analysis of the two algorithms. b) Applying DFS considering 'A' as the starting vertex , check whether the graph given in fig 2b is strongly connected or not, if not strongly connected then find the strongly connected components.	05+05
			OR	
3.	CO2 CO3	L2, L3 2.6.4, 2.5.3	a) Design an efficient algorithm to find the number of substrings that starts with some character 'X' and ends with ' Y'(for example: there are 4 such substrings in ABAAEBDA which starts with A and ends with B) and Analyze its efficiency. b) Traverse the graph given in fig 3b in BFS showing the contents of the data structure used and also construct the forest.	05+05

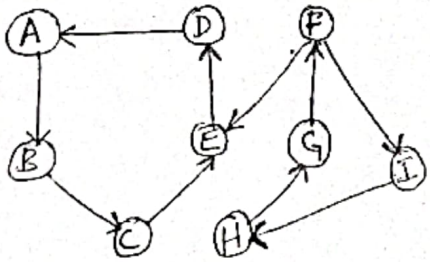

Fig 2b

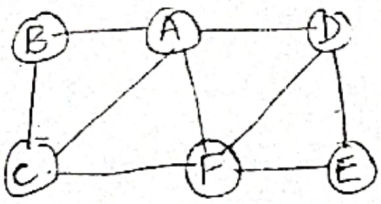

Fig 3b

USN											

"20CS420"

JSS MAHAVIDYAPEETHA
JSS SCIENCE AND TECHNOLOGY UNIVERSITY, MYSURU

Computer Science and Engineering
IV Semester: CIE - 5 (Event-V)

Design and Analysis of Algorithms

Duration:1 Hr. Date: 07.07.2023 Max. Marks:20

NOTE: PART-A is Compulsory, PART-B has internal choice

PART-A

Q.NO	CO	CD/PI	QUESTION	MARKS
1.	CO3 CO4	L2 2.5.2	a) Outline an algorithm to compute Mode in a list of elements with presorting and perform its analysis. b) For the graph shown in fig1.b, find the shortest path between every pair of vertices using dynamic programming technique.	5+5

PART-B

2.	CO3 CO4	L2, L3 2.5.3	a) Sort the list {'M','E',R','G','E','S','O','R','T'} in alphabetical order by constructing Max Heap. Show the Heap contents after each deletion. b) Compute C(6,5) with dynamic programming technique and compare the growth of additions while computing C(n,k) with dynamic programming and directly solving with recursion without dynamic programming.	05+05
			OR	
3.	CO3 CO4	L2, L3 2.6.4, 2.5.3	a) What are AVL trees? Illustrate with examples and general forms the different rotations used in AVL trees. b) Discuss with an example, the general idea of Dynamic Programming technique. How this is similar to and different from divide and conquer technique? Illustrate with an example. OR Construct 2-3 tree for the list {'R','E','P','R','E','S','E','N','T'} and prove that the time efficiency of searching, insertion and deletion in 2-3 trees are all in $\Theta(\log n)$ in the worst case.	05+05

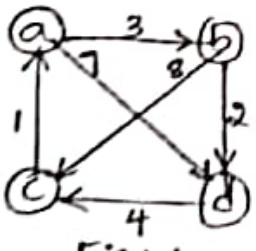

Fig 1.b

JSS MAHAVIDYAPEETHA

JSS SCIENCE AND TECHNOLOGY UNIVERSITY, MYSURU

IV Semester BE Degree Examination

Department of Computer Science and Engineering

DESIGN AND ANALYSIS OF ALGORITHMS

Duration: 3 Hours | Max. Marks: 100

NOTE: Part-A is compulsory and Part-B has internal choice

PART – A

Q.NO	CO	CD	PI	QUESTION	MARKS
1.	CO1	L1	2.5.2	Discuss with neat diagram, sequence of steps to be followed while designing and analyzing algorithms.	10
2(a).	CO1	L1	1.7.1	Discuss the strengths and weakness of brute force design strategy.	4
2(b).	CO3	L4	4.4.2	Design an efficient polynomial evaluation algorithm with brute force design technique and analyze the same to prove its linear efficiency class.	6
3(a).	CO2	L3	4.4.2	Discuss how presorting can be used to improve the efficiency of brute force element uniqueness algorithm and mode computation algorithm.	5
3(b).	CO4	L4	2.6.4	Design an algorithm to compute a^n using decrease and conquer design technique and analyze its efficiency.	5
4.	CO3	L4	2.5.2	Discuss how Dynamic programming is different from divide and conquer technique. Design a dynamic programming algorithm for computing C(n,k). Analyze its space and time complexity. Trace the same for C(6,3).	10
5.	CO2	L3	1.7.1	Discuss how decision trees are useful in establishing the lower bound of an algorithm. Illustrate the same to establish the lower bound of comparison based sorting algorithms with example	10

PART – B

Q.NO	CO	CD	PI	QUESTION	MARKS
6(a).	CO3	L4	2.5.3	Design an iterative algorithm to find the sum of square of first 'n' natural numbers. Analyze the same to establish the order of growth of the algorithm's basic operation count. Is it possible to improve the efficiency of your algorithm? Justify your answer.	5

6(b).	CO3	L4	10.4.1	For the algorithm given below answer the following.
				xyz(A[0…..n-1,0…..n-1])
				for i ←0 to n-2 do
				for j ← i+1 to n-1 do
				if A[i,j] ≠ A[j,i]
				return false
				return true
				i) What does this algorithm compute?
				ii) Establish the order of growth for the algorithm's basic operation count.
				iii) Is it possible to improve the efficiency of this algorithm? Justify your answer.
					5

				OR	
7.	CO1	L2	1.7.1	Discuss with examples the asymptotic notations used to compare and rank the order of growth of algorithms basic operation count.	10
8.	CO4	L4	2.5.3	Design a recursive merge sort algorithm, specify the worst case and best case input and analyze the same to establish the order of growth. Trace the same in sorting the string Q,U,I,C,K,S,O,R,T.	10
				OR	
9.	CO4	L4	2.5.3	Design a recursive Quick sort algorithm, specify the worst case and best case inputs and analyze the same. Trace the same in sorting the string M,E,R,G,E,S,O,R,T. Draw the tree of recursive calls.	10

10(a)	CO3	L2	3.6.2	For the graph shown in Figure 10(a), starting from the vertex 'a' construct Breadth first search forest and give the order in which the vertices are visited showing the contents of the data structure used.	4
10(b)	CO3	L4	3.6.2	sort the list H,E,A,P,S,O,R,T in alphabetical order by applying heap sort. Analyze its efficiency.	6
				OR	
11(a).	CO2	L4	2.5.3	Design distribution counting algorithm to sort an array. Analyze its efficiency.	5
11(b).	CO2	L3	2.5.3	Construct AVL tree and 2-3 tree for the following list of values, 5,6,8,3,2,4,7	5
12(a).	CO4	L3	2.5.2	Apply Prim's algorithm to find minimum cost spanning for the graph in fig.12(a). Include in the priority queue all the vertices not already in the tree .Choose vertex 'a' as the starting vertex.	5
12(b).	CO2	L3	2.5.2	Solve the instance in fig.12(b) to find the single source shortest paths with vertex 'a' as the source.	5
				OR	
13.	CO3	L3	1.7.1	Discuss how knapsack problem can be solved using dynamic programming with bottom-up approach. Apply the same to solve the following instances, find the composition and the Analyze its efficiency. Capacity W = 6	10

Item	Weight	Value
1	3	$25
2	2	$20
3	1	$15
4	4	$40
5	5	$50

14(a).	CO3	L3	4.4.2	Apply Backtracking to solve the following instance of subset sum problem S={1,3,4,5} d=10. Draw the state space tree.	5

<table>
<tr><td>14(b).</td><td>CO1</td><td>L2</td><td>2.6.4</td><td>Appraise the use of Backtracking and Branch and bound techniques.</td><td>5</td></tr>
<tr><td colspan="6">OR</td></tr>
<tr><td>15.</td><td>CO4</td><td>L3</td><td>4.4.2</td><td>Solve the assignment problem for the following instance and obtain the optimal solution. Draw the state space tree for the same.
<table>
<tr><td></td><td>J1</td><td>J2</td><td>J3</td><td>J4</td></tr>
<tr><td>P1</td><td>9</td><td>2</td><td>7</td><td>8</td></tr>
<tr><td>P2</td><td>6</td><td>4</td><td>3</td><td>7</td></tr>
<tr><td>P3</td><td>5</td><td>8</td><td>1</td><td>8</td></tr>
<tr><td>P4</td><td>7</td><td>6</td><td>9</td><td>4</td></tr>
</table></td><td>10</td></tr>
</table>

Course Outcome: At the end of the course the students will have the ability to	
CO–1	Understand various algorithm design techniques and Mathematical models.
CO–2	Apply appropriate data structures and suitable design technique to develop an algorithm for the given problem.
CO–3	Analyse the problem domain, use mathematical analysis model to estimate the running time efficiency of different algorithms for best, average and worst cases.
CO–4	Evaluate a suitable design technique to develop an algorithm for the given problem.

Performance Indicator:

1.7.1	Apply theory and principles of computer science and engineering to solve an engineering problem.
2.5.2	Identify processes/modules/algorithms of a computer based system and parameters to solve a problem.
2.5.3	Identify mathematical algorithmic knowledge that applies to a given problem.
2.6.4	Compare and contrast alternative solution/methods to select the best methods.
3.6.2	Able to produce a variety of potential design solutions suited to meet functional requirements.
4.4.2	Able to choose appropriate procedure/algorithm, dataset and test cases.