

QP Code:

Reg. No

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

SAVEETHA ENGINEERING COLLEGE

(Autonomous, Affiliated to Anna University, Chennai)

B.E./B.Tech. End Semester Examinations – Apr/May 2025 (R2019/R2024)

Common to Computer Science and Engineering / Information Technology

19CS402– Design and Analysis of Algorithms

Time: Three hours

Maximum marks: 100

Faculty Name	MANICKAM S	Department	COMPUTER SCIENCE ENGINEERING
---------------------	-------------------	-------------------	---

Answer All Questions

PART A

(10 x 2 = 20 marks)

Q.NO	Questions	CO	Knowledge Level (Blooms)	Difficulty Level (1-5)
QA101 *	Define the big oh notations with its properties.	CO1	K2	3
QA102 *	Why is constant time complexity denoted as O(1)?	CO1	K2	3
QA103	What is time complexity of the function fun() int fun(int n) { int count = 0; for (int i = n; i > 0; i /= 2) for (int j = 0; j < i; j*=2) count += 1; return count; }	CO1	K2	3
QA104	Examine average, best-case and worst-case efficiency of an algorithm.	CO1	K2	2
QA105	Outline the general plan for analyzing recursive algorithms.	CO1	K2	3
QA106	What is the output of the following code? void my_recursive_function(int n) { if(n == 0) return; printf("%d ",n); my recursive function(n-1);	CO1	K2	4

	<pre> } int main() { my_recursive_function(5); return 0; } </pre> <p>Justify your answer with a recursion tree.</p>			
QA107	Apply the common technique for proving the correctness of an algorithm.	CO1	K2	2
QA108	<p>Consider the following function f:</p> <pre> int f(int n) { int s = 0; while(n > 1) { n = n/2; s++; } return s; } </pre> <p>What is the asymptotic complexity in terms of n?</p>	CO1	K3	4
QA201 *	How does the divide and conquer technique apply to searching algorithms	CO2	K2	3
QA202	<p>Examine the average number of key comparisons required for a successful search for sequential Search on n items and estimate its best time complexity for the comparison.</p>	CO2	K3	3
QA203	List out the difference between Linear technique from binary search technique.	CO2	K2	2
QA204 *	Discuss the processing steps in Quick sort.	CO2	K2	2
QA205	Determine the Time complexity and drawback of Quick sort algorithm.	CO2	K3	3
QA206	<p>Give an example of a text of length n and a pattern of length m that constitutes a worst-case input for the brute force string matching algorithm. Formulate and find how many character comparisons will be made for such input.</p> <p>For Example, Text = abcabcabb, Pattern = red</p>	CO2	K3	3
QA207	Analyze the Time efficiency and drawback of Quick sort algorithm.	CO2	K3	3
QA208	Examine a brute force algorithm for counting the number of vowels in a given text.	CO2	K3	3
QA301 *	Difference between Dynamic programming and Backtracking.	CO3	K3	2
QA302	Illustrate the concept of cherry pickup problem with an example.	CO3	K3	2
QA303	Compare greedy approaches with dynamic programming.	CO3	K3	4
QA304	<p>Consider the following array: {1, 3, 5, 8, 9, 2, 6, 7, 6}</p> <p>What is the minimum number of jumps required to reach the end of the array?</p>	CO3	K3	3
QA305	Interpret and find the longest increasing subsequence for the given sequence: {10, -10, 12, 9, 10, 15, 13, 14}.	CO3	K2	2

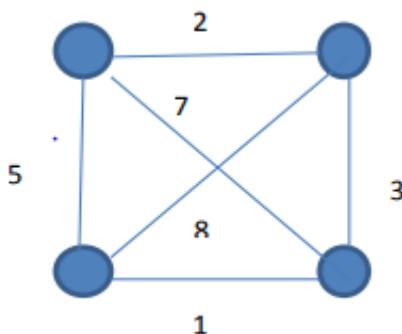
QA306 *	Illustrate the concept of graph coloring problem with an example.	CO3	K3	2
QA307	Show an algorithm to make for 1655 using the dynamic programming strategy. The coins available are {1000, 500, 100, 50, 20, 10, 5}.	CO3	K3	2
QA308	<p>In the given graph</p> <p>How many intermediate vertices are required to travel from node a to node e at a minimum cost?</p>	CO3	K3	2
QA401 *	Explore the principle of backtracking with branch and bound.	CO4	K2	2
QA402	Illustrate the term state space tree with example.	CO4	K3	3
QA403	Define bipartite graphs.	CO4	K3	2
QA404	Determine the chromatic number of the following graph. 	CO4	K2	3
QA405	Outline the various pattern matching algorithms.	CO4	K2	2
QA406	Describe the various pattern matching algorithms.	CO4	K2	2
QA407	Draw a state space tree for the following subset problem for the values { 3,5,6,7} with the target sum 15.	CO4	K3	2
QA408 *	Advantages of branch and bound algorithm over backtracking?	C04	K3	2
QA501 *	State Euler Circuit problem.	CO5	K2	3
QA502	How is the accuracy of the approximation algorithm measured?	CO5	K2	3
QA503 *	Differentiate Euler circuit and Hamiltonian circuit problem?	CO5	K3	3
QA504	You are given a knapsack that can carry a maximum [weight of 60]. There are 4 items with weights {20, 30, 40, 70} and values {70, 80, 90, 200}. What is the maximum value of the items you can carry using the knapsack?	CO5	K3	3
QA505	State the solution for N-Queen Problem	CO5	K2	2
QA506	Describe the concept of Traveling Salesman Problem	CO5	K2	2
QA507	Differentiate between Polynomial and Non-Polynomial.	CO5	K2	3
QA508	Outline the general plan of exhaustive search.	CO5	K2	2

Q.NO		Questions	CO	Knowledge Level (Blooms)	Difficulty Level (1-5)
QB101	(a)*	Elaborate on the core principles and methodologies involved in algorithmic problem-solving (OR)	CO1	K2	3
QB101	(b)	Analyze the recursive version of the factorial function. Derive the recurrence relation for the same.	CO1	K2	3
QB102	(a)	Elaborate the most important problem types are used to illustrate different algorithm design techniques and methods of algorithm analysis. (OR)	CO1	K2	2
QB102	(b)	Analyze the recursive version to find the GCD of two numbers. Derive the recurrence relation for the same.	CO1	K4	4
QB103	(a)*	Write an algorithm to check whether the given element present in the given array using linear search. Find the worst-case complexity of the same. (OR)	CO1	K3	4
QB103	(b)*	Analyze and write an algorithm to sort a given list of elements using selection sort. Show the operation of the algorithm, on the list 10, 92, 38, 74, 56, 19, 82, 37 and estimate its time complexity.	CO1	K2	4
QB104	(a)	Explain the forward substitution and backward substitution method for solving recurrence relation with examples. (OR)	CO1	K2	4
QB104	(b)	Analyze the recursive version of the Fibonacci function. Derive the recurrence relation for the same.	CO1	K4	4
QB201	(a)*	Write a Python program for Merge sort. Analyze the algorithm for the best case, average case and worst case. (OR)	CO2	K3	3
QB201	(b)	Write the KMP string matching algorithm for finding a pattern on a text, and analyze the algorithm.	CO2	K2	3
QB202	(a)	Create a Python program to perform linear search on a sorted list of elements. (OR)	CO2	K2	2

		Write a Python program to implement pattern matching on the given string using Brute Force algorithm.	CO2	K2	2
QB202	(b)				
		Develop a Python program to perform binary search on a sorted list of elements using the iterative approach.	CO2	K4	3
QB203	(a)	(OR)			
		Write a Python Program to find minimum number of swaps required to sort an array given by the user	CO2	K3	4
QB203	(b)				
		Analyze and write an algorithm to sort a given list of elements using merge sort. Show the operation of the algorithm, on the list 38, 27, 43, 3, 9, 82, 10. Predict the complete analysis for the same.	CO2	K4	3
QB204	(a)	(OR)			
		Write a python program using quick sort to sort the given list of values.	CO2	K3	4
QB204	(b)*	Write a Python program for Merge sort. Analyze the algorithm for the best case, average case and worst case.	CO2	K3	3
QB301	(a)	Demonstrate the methodology for the longest common subsequence using dynamic programming and develop the Python code for longest common subsequence.	CO3	K3	4
		(OR)			
QB301	(b)	i) Explain kadane's algorithm and its time complexity (7 marks) ii) Write a Python program using kadane's algorithm (6 marks)	CO3	K4	4
QB302	(a)	Create a Python function to find the minimum number of jumps needed to reach the end of the array using Dynamic Programming.	CO3	K4	4
		(OR)			
QB302	(b)	Create a Python program to find longest common substring or subword (LCW) of two strings using dynamic programming with top-down approach or memoization.	CO3	K3	3
QB303	(a)*	Write an algorithm to the longest common substring problem. Explain the methodology with a python code.	CO3	K3	4

		(OR)																		
QB303	(b)	Create a Python function to compute the fewest number of coins that we need to make up the amount given for coin change problem using dynamic programming	CO3	K3	4															
QB304	(a)*	Give the pseudo code for Prim's algorithm and apply the same to find the minimum spanning tree of the graph shown below.	CO3	K3	4															
		(OR)																		
QB304	(b)	Estimate the following instance of the knapsack by branch and bound algorithm. Knapsack Capacity W = 10.	CO3	K3	4															
		<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Item</th> <th>Weight</th> <th>Values</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>4</td> <td>\$40</td> </tr> <tr> <td>2</td> <td>7</td> <td>\$42</td> </tr> <tr> <td>3</td> <td>5</td> <td>\$25</td> </tr> <tr> <td>4</td> <td>3</td> <td>\$12</td> </tr> </tbody> </table>	Item	Weight	Values	1	4	\$40	2	7	\$42	3	5	\$25	4	3	\$12			
Item	Weight	Values																		
1	4	\$40																		
2	7	\$42																		
3	5	\$25																		
4	3	\$12																		
QB401	(a)*	Illustrate the procedure to solve n-queens problem using backtracking approach.	CO4	K3	3															
		(OR)																		
QB401	(b)	Build the Python code for Hamiltonian circuit problem using back-tracking	CO4	K3	4															
QB402	(a)	i)Create a Python program to find Minimum number of jumps to reach end of the array using naive method(recursion) (6 marks)	CO4	K2	3															

		ii)Explain the different methods used to solve the problem Minimum number of jumps to reach end of the array (7 marks)			
		(OR)			
QB402	(b)*	i)Explain the algorithms used in pattern matching(7 marks) ii)Develop a Python program to implement pattern matching on the given string.(6 marks)	CO4	K4	4
QB403	(a)	Demonstrate Knight Tour Problem using backtracking and write Python function for knight tour problem.	CO4	K4	4
		(OR)			
QB403	(b)	State the subset-sum problem and write a Python program to find the count of the subset that matches the target sum.	CO4	K2	3
QB404	(a)	Explain how to find the Minimum cost path in a Directed Graph with examples and algorithms.	CO4	K4	4
		(OR)			
QB404	(b)*	i)Write a python program to check whether the Hamiltonian path exists in the given graph (9 marks). ii)List some of the applications of Hamiltonian Circuit Problem (4 marks).	CO4	K2	3
QB501	(a)	Explain how to solve Sudoku Solver using the concept of backtracking .	CO5	K2	3
		(Or)			
QB501	(b)	Describe an algorithm for the subset sum problem. Let w= {5, 7, 10, 12, 15, 18, 20} and m=35. Create the Python program to display the subsets equal to the given sum.	CO5	K2	3
QB502	(a)	Define Class NP. Discuss about any five problems for which no polynomial time algorithm has been found.	CO6	K2	3
		(Or)			
QB502	(b)	Create a Python program for 0/1 knapsack problem using naive recursion method.	CO5	K2	3
QB503	(a)	Solve traveling salesman problem for the given graph .	CO5	K3	3



(Or)

QB503	(b)*	Analyze the approximation algorithm for travelling salesperson problem with example.	CO5	K3	3												
QB504	(a)	<p>Find the optimal solution to the knapsack problem with given data.</p> <table style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Item</th> <th>Weight</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>15</td> <td>60</td> </tr> <tr> <td>B</td> <td>20</td> <td>100</td> </tr> <tr> <td>C</td> <td>30</td> <td>120</td> </tr> </tbody> </table>	Item	Weight	Value	A	15	60	B	20	100	C	30	120	CO5	K3	4
Item	Weight	Value															
A	15	60															
B	20	100															
C	30	120															
QB504	(b)*	(Or) Write a naive pattern matching algorithm to compare a specified pattern with a given text string.	CO5	K3	3												

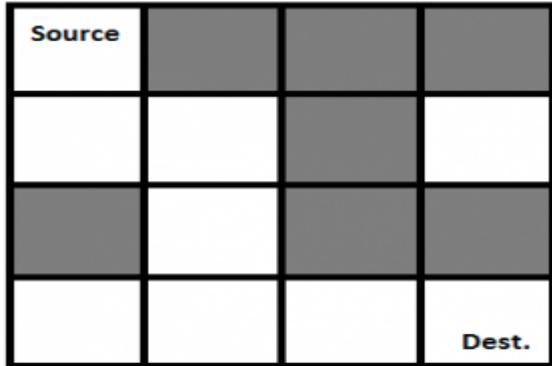
PART C

(1 x 15 = 15 marks)

(Case study/Comprehensive type Questions)

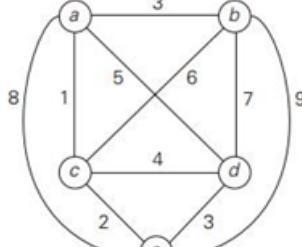
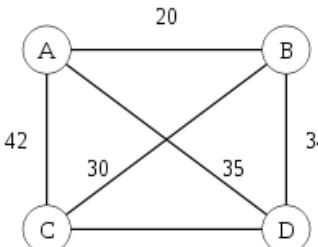
Q.NO		Questions	CO	Knowledge Level (Blooms)	Difficulty Level (1-5)
QC101	(a)*	Analyze the algorithm and write a python program to calculate the sum of the series $1+2+3+\dots N$ using Recursive function and estimate its category of running time.	CO1	K4	3
		(OR)			
QC101	(b)	<p>Infer if the following equalities are correct.</p> <p>a) $n^3 + 20n + 1 = O(n^3)$.</p> <p>b) $n^3 + 20n + 1 \neq O(n^2)$.</p> <p>c) $n^3 + 20n = \Omega(n^2)$.</p> <p>d) $5n^2 - 6n = \Theta(n^2)$.</p>	CO1	K4	3

QC201	(a)*	Write a Python program for the implementation of merge sort on the given list of values. Analyze the Time and Space Complexity with Quick sort.	CO2	K3	4
		(OR)			
QC201	(b)	Write a Python function to implement Quick Sort on a given list of values. The function should print the sorted list and the pivot value at each iteration. And, discuss the worst-case time complexity of the Quick Sort algorithm.	CO2	K3	4
QC301	(a)*	Create a recursive python program to find the minimum number of operations to convert str1 to str2 using brute force approach. input: str1=python str2=peithen output: Edit distance 3	CO3	K4	4
		(OR)			
QC301	(b)	i)Explain cherry pickup problem and its time complexity (7 marks). ii)Implement a Cherry Pickup problem using Python. (6 marks).	CO3	K4	5
QC401	(a)	Explain Kadane's algorithm for finding the maximum sum of a contiguous subarray within an integer array. Provide a step-by-step explanation of the algorithm and illustrate it with a Python code.	CO4	K4	4
		(OR)			
QC401	(b)*	Rat In a Maze Problem. You are given a maze in the form of a matrix of size n * n. Each cell is either clear or blocked denoted by 1 and 0 respectively. A rat sits at the top-left cell and there exists a block of cheese at the bottom-right cell. Both these cells are guaranteed to be clear. You need to find if the rat can get the cheese if it can move only in one of the two directions - down and right. It can't move to blocked cells.	CO4	K4	4



Provide the solution for the above problem Consider n = 4

The output (Solution matrix) must be 4*4 matrix with value "1" which indicates the path to destination and "0" for the cell indicating the absence of the path to destination. Write a python function to solve rat in maze problem.

QC501	(a)	Solve the following traveling salesperson problem using Branch and Bound algorithm. 	CO5	K4	4
		(OR)			
QC501	(b)	Solve the following travelling salesperson problem using Branch and Bound algorithm. 	CO5	K4	4

Knowledge Level (Blooms Taxonomy)					
K1	Remembering (Knowledge)	K2	Understanding (Comprehension)	K3	Applying (Application of Knowledge)
K4	Analysing (Analysis)	K5	Evaluating (Evaluation)	K6	Creating (Synthesis)

General Instructions

- (i) For each Question, mention K1 or K2 etc. for Knowledge Level**
- (ii) For each Question, mention CO1, CO2 etc. for Course Outcomes.**
Verify the COs with the Syllabus before framing the Questions.
An Either or type Question should have the same CO in both (a) and (b) parts.
- (iii) For each Question, mention any number from 1 to 5 for Difficulty Level**
(With 1 as Most Easy & 5 as Most Difficult)
- (iv) Mark with * near the Q.No for those Questions which are framed newly and were not included in the QRs of LAST TWO SEMESTERs. Ensure minimum 2 new updatations per unit in all Parts.**
- (v) Allot split up of marks for subdivisions.**
- (vi) DO NOT Copy and Paste Equations as Images.**
All Mathematical Equations should be typed using the appropriate tools.
- (vii) Type the Answer Key in the same QR template across the respective Q.Nos and delete the last 3 columns (CO, KL, Diff level) in the Answer Key file....**