ii. You are planning a backpacking trip and you have a limited capacity backpack that can carry a maximum weight of W kilograms. You want to carry some essential and indivisible items with you, each with its own weight and value. Your goal is to maximize the total value of the items you can carry in your backpack without exceeding its weight capacity. **6**

Items:
Tent: Weight - 3 kg, Value - $150
Sleeping Bag: Weight - 2 kg, Value - $100
Cooking Stove: Weight - 1.5 kg, Value - $80
Water Purifier: Weight - 1 kg, Value - $50
Food Supplies (for the entire trip): Weight - 5 kg, Value - $200
First Aid Kit: Weight - 0.5 kg, Value - $30
Backpack Capacity: W = 8 kg

Using the 0/1 Knapsack problem solving technique, determine the optimal selection of items to carry in your backpack to maximize the total value. Provide a list of selected items and the total value that can be carried within the given weight capacity.

OR iii. Consider two sequences of characters: Sequence A with elements {A, B, C, D, E, F} and Sequence B with elements {B, D, F, G}. Determine the Longest Common Subsequence (LCS) of these sequences using dynamic programming. Provide the LCS and explain the steps you took to find it. **6**

Q.6 Attempt any two:

i. Provide a step-by-step explanation of the branch and bound algorithms, highlighting its key components and strategies. Illustrate your explanation with a suitable example, demonstrating how this algorithm design technique can be applied to solve a specific optimization problem. **5**

ii. Discuss the graph coloring problem in the context of real-world applications. Provide at least two examples of practical scenarios where graph coloring is used as a problem-solving technique. Describe how graph coloring algorithms can be employed to solve these real-world problems effectively. **5**

iii. Compare and contrast NP-Complete problems and NP-Hard problems. Provide examples of problems that belong to each of these classes and explain why they are categorized as NP-Complete or NP-Hard. **5**

******

Total No. of Questions: 6 *Total No. of Printed Pages:4*

Enrollment No......................................

Faculty of Engineering

End Sem Examination Dec-2023

CS3CO42 Design & Analysis of Algorithms

Programme: B.Tech. Branch/Specialisation: CSE All

Duration: 3 Hrs. **Maximum Marks: 60**

Note: All questions are compulsory. Internal choices, if any, are indicated. Answers of Q.1 (MCQs) should be written in full instead of only a, b, c or d. Assume suitable data if necessary. Notations and symbols have their usual meaning.

Q.1 i. In asymptotic notation, which notation describes the lower bound of an algorithm's running time? **1**
(a) Θ-notation (b) O-notation (c) Ω-notation (d) β-notation

ii. Which of the following quadratic sorting techniques has the worst-case time complexity of O(n^2), making it inefficient for large datasets? **1**
(a) Merge Sort (b) Quick Sort
(c) Heap Sort (d) None of these

iii. In the divide and conquer sorting technique, which algorithm uses a pivot element to partition the array into two subarrays and then recursively sorts those subarrays? **1**
(a) Bubble Sort (b) Quick Sort
(c) Merge Sort (d) Insertion Sort

iv. What is the time complexity of Strassen's matrix multiplication algorithm for multiplying two n × n matrices? **1**
(a) O(n^2.81) (b) O(n^2)
(c) O(n^3) (d) O(n log n)

v. In Huffman coding, what is the fundamental principle used to encode data efficiently? **1**
(a) Arithmetic coding
(b) Run-length encoding
(c) Burrows-wheeler transform
(d) Prefix coding

vi. What is the time complexity of Dijkstra's algorithm when implemented with a binary heap for priority queue? **1**
(a) O(V log V) (b) O((V + E) log V)
(c) O(E + V log V) (d) O(V^2)

vii. What would be the running time of the dynamic programming approach to solve the Longest Common Subsequence (LCS) problem for two strings of lengths *m* and *n* respectively? **1**

(a) O(m×n) (b) $O(2^{m+n})$

(c) O(m+n) (d) O(m×n×min (m,n))

viii. In dynamic programming, what is the key feature that distinguishes it from other algorithmic techniques like divide and conquer or greedy algorithms? **1**

(a) Randomization (b) Recursion (c) Memorization (d) Backtracking

ix. In the context of the N-Queens problem, what is the maximum number of queens that can be placed on an N×N chessboard such that no two queens threaten each other? **1**

(a) N (b) 2N−1 (c) N^2 (d) $2^N - 1$

x. In computational complexity theory, a problem is NP-complete when: **1**

(a) It is a decision problem, meaning that for any input to the problem, the output is either "yes" or "no".

(b) The correctness of each solution can be verified quickly (namely, in polynomial time) and a brute-force search algorithm can find a solution by trying all possible solutions.

(c) Both (a) & (b)

(d) None of these

Q.2 i. What is the master theorem? How is it used to solve recurrence relations? **2**

ii. Explain the differences between Big O, Omega, and Theta notation in the context of asymptotic analysis. Provide examples for each notation. **3**

iii. Consider a modified version of the Insertion Sort algorithm called "Reverse Insertion Sort." In this variation, instead of sorting the array in ascending order, the algorithm sorts the array in descending order. Explain the step-by-step process of Reverse Insertion Sort with the help of a detailed example. Also, discuss the time complexity of this modified algorithm and compare it with the standard Insertion Sort algorithm. **5**

OR iv. Explain the bubble sort algorithm in detail, covering its working principles, time complexity analysis, and an example demonstrating the sorting process. Also, discuss its advantages and limitations in practical applications. **5**

Q.3 i. Explain the concept of a "bucket" in the radix sort algorithm and how it is used to sort elements. **2**

ii. Describe the steps involved in the heap sort algorithm. Include the processes of building the initial max heap, swapping elements, and maintaining the heap property. Provide the algorithmic implementation and analyze its time complexity. **8**

OR iii. Consider an unsorted array of integers that needs to be sorted using the quicksort algorithm. The array to be sorted is {5, 2, 9, 1, 5, 6, 13, 4, 21, 10}. Apply the QuickSort algorithm to sort the given array step by step, showing the partitioning process at each step. Calculate the time complexity of the QuickSort algorithm for the given array considering the number of comparisons and swaps made during the sorting process. **8**

Q.4 i. Explicate the concept of greedy technique in algorithm design and clarify when it is appropriate to use greedy techniques in problem-solving. Provide reasoning for your answer and illustrate your explanation with a relevant example. **3**

ii. You are given a set of characters and their frequencies: **7**

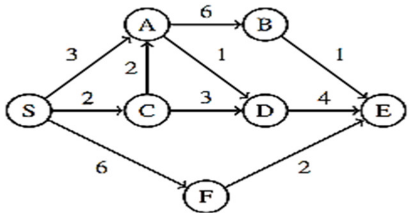
Character	Frequency
A	5
B	9
C	12
D	13
E	16
F	45

Show the step-by-step process of constructing the Huffman tree, including merging nodes and updating frequencies.

Calculate the Huffman codes for each character in the final tree.

Calculate the average encoding length of the characters in the resulting Huffman tree.

OR iii. Consider the following graph representation: **7**



Apply Dijkstra's algorithm to find the shortest paths from the source node S to all other nodes in the graph. Show the step-by-step process, including the initialization, updating of distances, and selection of nodes. Provide the final shortest paths and their respective distances.

Q.5 i. Explain the steps involved in multistage graph in dynamic programming and provide suitable example to illustrate each step. **4**

P.T.O.

Marking Scheme
Design and Analysis of Algorithm-CS3CO42(T)

Q.1	i)	(c) Ω-notation		**1**
	ii)	(d) None of these		**1**
	iii)	(b) Quick Sort		**1**
	iv)	(a) O(n^2.585)		**1**
	v)	(d) Prefix coding		**1**
	vi)	(b) O((V + E) log V)		**1**
	vii)	(a) O(m×n)		**1**
	viii)	(c) Memoization		**1**
	ix)	(a) N		**1**
	x)	(c) Both (a) and (b) are correct5		**1**
Q.2	i.	For Master Theorem -	1 Mark	**2**
		Solving ………theorem-	1 Mark	
	ii.	Big O- Explanation with example	1 Mark	**3**
		Omega Explanation with example	1 Mark	
		Theta Explanation with example -	1 Mark	
	iii.	Time Complexity	1 Mark	**5**
OR	iv.	Bubble sort …… example –	4 Marks	**5**
		Advantages and practical application	1 Mark	
Q.3	i.	The concept of a "….. elements.	(As per explanation)	**2**
	ii.	Heap sort including building of heap –	5 Marks	**8**
		Algorithm writing –	2 Marks	
		Time complexity analysis -	1 Marks	
OR	iii.	Quick sort illustration –	5 Marks	**8**
		Algorithm writing -	2 Marks	
		Complexity analysis-	1 Mark	
Q.4	i.	Greedy concept and clarification-	2 Marks	**3**
		Reasoning with example -	1 Mark	
	ii.	Huffman tree constructiob -	4 Marks	**7**
		Calculating Huffman codes -	2 Marks	
		Calculating average encoding -	1 Marks	
OR	iii.	Process	5 Marks	**7**
		Shortest Paths and distances	2 Marks	
Q.5	i.	Steps of DP-	2 Marks	**4**
		Example of each step -	2 Marks	
	ii.	Process	5 Marks	**6**
		List of selected item (Solution)	1 Mark	
OR	iii.	Process	5 Marks	**6**
		Final Answer	1 Mark	
Q.6				
	i.	Step	2 Marks	**5**
		Example	1 Mark	
		Demonstration	2 Marks	
	ii.	Discussion	2 Marks	**5**
		Example	1 Mark	
		Description	2 Marks	
	iii.	Comparison	2 Marks	**5**
		Example	1 Mark	
		Explanation	1 Mark	

******