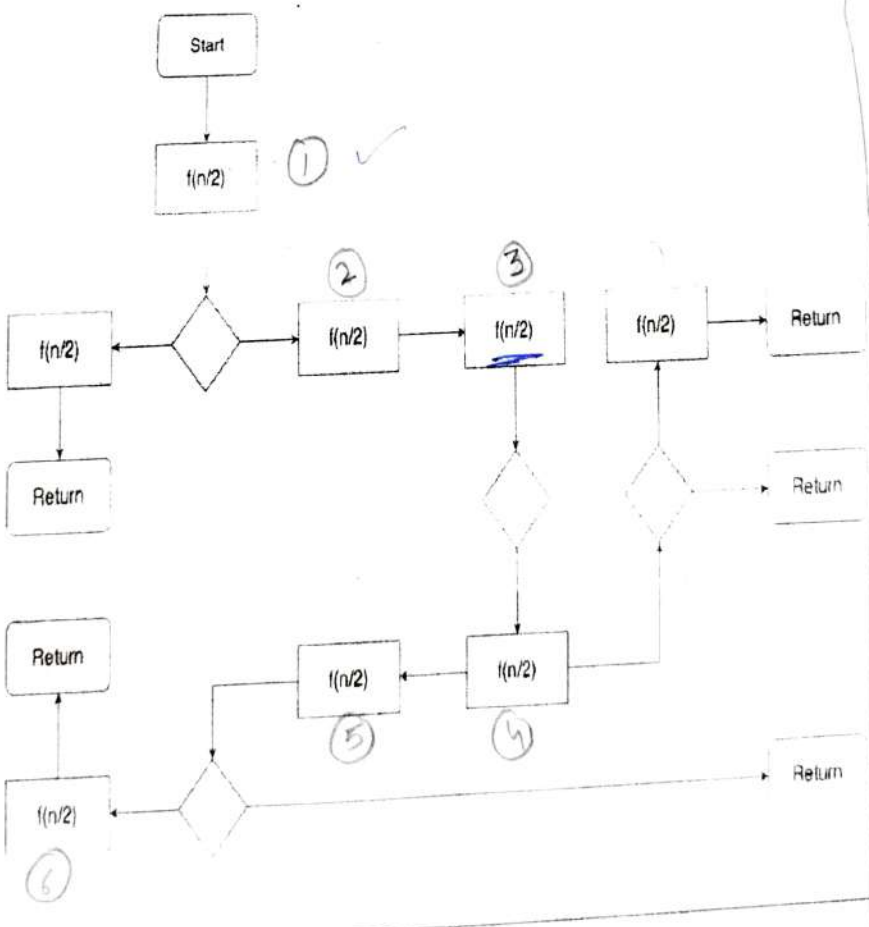
Degree: B. TECH Semester: III[rd]

MID-SEMESTER EXAMINATION, September 2023

Max Marks: 15
Duration: 1:30 Hours

Course Title: Design and Analysis of Algorithms
Course Code: COCSC06/ CACSC06/CDCSC06 CMCSC06

Note: Attempt all questions in given order only. Missing data/information (if any) may be suitably assumed & mentioned in the answer.

Q. No.	Question	Marks	CO
1a.	Let **A** be a unsorted array of **n** integers. Devise the most efficient algorithm which determine whether two elements whose sum is less than 10^6 exists in **A** or not. Moreover, compute asymptotic tight bound of time complexity for the devised algorithm.	1.5	CO5
1b.	The following diagram shows the flowchart for a recursive function **f(n)**. Assume that all statements, except for the recursive calls, have **O(1)** time complexity. If the worst case time complexity of this function is $\mathbf{O(n^{\alpha})}$, then compute the least possible value (accurate up to two decimal positions) of α. 	1.5	CO1
2a.	1. Consider the following c function: **double** *foo* (**int** n)	1.5	CO1

	{
	int i; –
	double sum; –
	if (n = = 0) **return** 1.0;
	else
	{
	sum = 0.0; –
	for (i = 0; i < n; i++) –
	sum += foo (i); –
	return sum; –
	}
	}
	Compute the time complexity and space complexity of above function.		

2b.	Suppose we modify the above function ***foo()*** and store the values of ***foo(i), 0 < = i < n***, as and when they are computed. After this modification in above program, what will the time and space complexity ***foo()*** ? Is there be any change in time and space complexity of this case concerning Q.2 (a).	1.5	CO1
3a.	Given the sequence of numbers 36, 96, 101, 29, 32, 55, 117, and 205, demonstrate how you would insert them into an empty red-black tree while maintaining the tree's properties.	1	CO3
3b.	Provide a pseudocode representation illustrating the heapify procedure for creating a max-heap from an array. Describe the essential steps involved in the procedure and discuss its time complexity.	2	CO3
4a.	How does the merge sort algorithm use the divide-and-conquer approach? Provide a brief pseudocode explanation of how it divides, conquers, and combines to achieve sorting. Discuss the time complexity of merge sort and highlight its stability and efficiency compared to other sorting algorithms -	2	CO4
4b.	How do you construct a B tree of order 3 from the following data 7, 2, 9, 11, 15, 5, and 37?	1	CO3
5a.	You are given the following list of integers to sort using Counting Sort and Radix Sort. Compare the time complexity of these two algorithms for this specific given data. Determine the number of steps/iteration for both the algorithms. Data: [235, 468, 129, 782, 932, 123, 634, 346, 349, 568]	2	CO5
5b.	Consider the following data set: [42, 19, 27, 14, 35, 10, 5, 30, 22, 15, 9, 12, 2, 45, 7]. Perform Shell Sort on this data set using the shell increments (gaps) of (7, 4, 2, 1). i). Perform the first step of the Shell Sort with a gap of 7 and show the intermediate result after this step. ii). Perform the second step of the Shell Sort with a gap of 4 and show the intermediate result after this step. iii). Calculate the exact number of comparisons made during above steps.	1	CO3

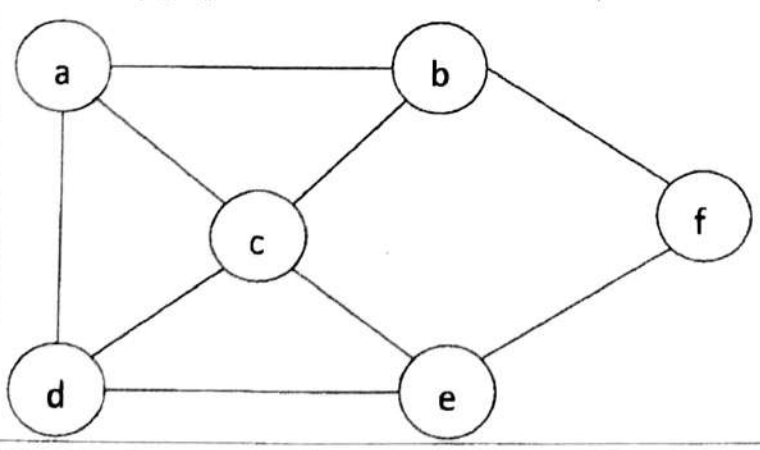
Degree: B. Tech Semester: 3rd - Course work
END-SEMESTER EXAMINATION, Nov-Dec 2023
Course Title: Design and Analysis of Algorithms
Course Code: COCSC06/ CACSC06/CDCSC06
CMCSC06

Time: 03 Hours | Max. Marks: 40

Note: - Attempt all the five questions. Missing data/ information (if any), maybe suitably assumed & mentioned in the answer.

	Question	Marks	CO
Q 1	Attempt any 2 parts of the following		
1a	Solve the recurrence relation with the help of master method, also show the process/reason: a. $T(n) = 3T(n/3) + \sqrt{n}$ b. $T(n) = 16T(n/4) + n!$ c. $T(n) = 16T(n/4) + n$	4	CO1
1b	Solve the recurrence relation using Recursion Tree, also present the number of leaf nodes, depth of three, cost at each level, and final cost: $T(n) = T(n/5) + T(4n/5) + n$	4	CO1
1c	Find the minimum number of comparisons needed to locate both the minimum and maximum values among 100 numbers, and elucidate the process involved.	4	CO5
Q 2	Attempt any 2 parts of the following		
2a	Create an AVL-tree for the following list of elements: 80, 40, 60, 20, 10, 30, 70, 50, 90, 100, 55, 35. Identify the number of rotations required for making it balanced tree.	4	CO3
2b	A. Derive the recurrence relations of the Best, Worst, and Average-case time complexities of the Quicksort algorithm and solve them to derive the asymptotic bounds in all cases. B. Give proof of correctness of insertion sort that it sorts its argument.	4	CO2
2c	What is the difference between linear sorting and comparison-based sorting? Explain the different steps of the counting sort algorithm with an example. Discuss its time complexity.	4	CO3
Q 3	Attempt any 2 parts of the following		
3a	A weighted graph indicating (non-negative) distances of roads between adjacent towns is given. A town planner wishes to identify the shortest set of roads to connect the towns. Devise an efficient algorithm, along with its complexity, that will enable the town planner to determine this.	4	CO4
3b	You are given an array ***A[1...n]*** representing a min-heap and an integer ***k***. You have to design an algorithm to output all the keys in the array that are less than ***k***. (For example, if ***k = 6*** and the array *A* is as shown below, then your algorithm should output the keys ***3*** and ***5***.)	4	CO5

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Key	3	10	5	13	17	6	11	15	16	21	18	9	8	23

	Give pseudocode for your algorithm and discuss running time. Let **m** be the number of keys in **A** that are smaller than **k**. The running time of your algorithm should be **O(m)**.		
3c	Solve the following instance of the 0-1 knapsack problem given the knapsack capacity in **W = 5** using dynamic programming and explain it. **Item** **Weight** **Cost** 1 4 10 2 3 20 3 2 15 4 5 25	4	CO4
Q 4	Attempt any 2 parts of the following		
4a	Assess the suitability of randomized algorithms for the following scenarios and provide explanations for each case: a) Merge Sort b) Minimum Spanning Tree c) Aircraft Control System d) Knapsack Problem	4	CO4
4b	Design a randomized algorithm to efficiently find the kth smallest element in an array containing distinct elements. Ensure that the algorithm achieves linear average case time complexity. Explain the algorithm's steps and analyse its time complexity.	4	CO5
4c	Elucidate the concept of randomized algorithms and their utility, using the Quick Sort algorithm as an example. Additionally, perform an analysis of the randomized Quick Sort algorithm, considering both its average and worst-case scenarios.	4	CO3
Q 5	Attempt any 2 parts of the following		
5a	Explain P, NP, NP-hard, and NP-complete class problems, give an example of each class of problems.	4	CO1
5b	Find the Hamiltonian cycle by using the backtracking approach for the following graph.  	4	CO3
5c	Explain 8-queens problem and apply backtracking to solve this problem.	4	CO4