Register No.: ..... Name: .....

SAINTGITS COLLEGE OF ENGINEERING (AUTONOMOUS)

(AFFILIATED TO APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY, THIRUVANANTHAPURAM)

SIXTH SEMESTER B.TECH DEGREE EXAMINATION (R), MAY 2023
COMPUTER SCIENCE AND ENGINEERING
(2020 SCHEME)

Course Code : 20CST306

Course Name: Algorithm Analysis and Design

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

PART A

(Answer all questions. Each question carries 3 marks)

1. Solve using Iteration method T(n)=2T(n/2)+n,T(1)=1.
2. Derive Average case running time of binary search.
3. Define AVL tree. What is the advantage of AVL tree? Give an example.
4. Discuss briefly the heuristics, union by rank and path compression, to improve the running time of disjoint set data structure.
5. Write the control abstraction of Greedy strategy.
6. Strassen's multiplication method is used to multiply two n x n matrices when n is a power of 2. How it can be modified when n is not a power of 2?
7. Discuss briefly the elements of dynamic programming with a suitable example.
8. Compare backtracking and branch-and-bound design techniques.
9. Define P, NP and NP complete domains.
10. Compare Las Vegas and Monte Carlo algorithms.

PART B

(Answer one full question from each module, each question carries 14 marks)

MODULE I

11. a) Define Big Oh, Big Omega and Theta notations and illustrate them graphically. (6)

    b) Solve the following recurrence using recursion tree method
    1) $T(n) = T(n/3) + T(2n/3) + cn$ (8)
    2) $T(n) = 2T(n/2) + n$

OR

12. a) Illustrate best case, average case and worst-case complexity with binary search algorithm. (9)

    b) Solve using Masters theorem
    i) $T(n)=2T(n/4)+\sqrt{n}$ (5)
    ii) $T(n)=7T(n/2)+ n^2$

MODULE II

13. a) Give Depth First Search algorithm for graph traversal. Perform its time complexity analysis. With example explain the algorithm. (9)
    b) Explain the advantages of using height Balanced Trees? Explain AVL Rotations. Give suitable example to explain the rotations. (5)

OR

14. a) Give Breadth First Search algorithm for graph traversal. Perform its complexity analysis. With example explain the algorithm. (9)
    b) With a graph explain the topological sorting and write the algorithm and perform its complexity. (5)

MODULE III

15. a) With an example, explain the single sources shortest path algorithm. Write the algorithm and perform its complexity. (7)
    b) Consider the following instance of Fractional Knapsack problem with 3 objects. The capacity of the knapsack is 20 units. The weights and profits of the 3 items respectively are represented by the vectors (w1,w2,w3) = (18,15,10) and (p1,p2,p3)=(25,24,15). Using a greedy strategy compute the optimal solution to this instance. (7)

OR

16. a) Illustrate the divide and conquer approach by applying 2 way merge sort for the input array: [15,12,14,17,11,13,12,16]. Write the recurrence for merge sort and give the complexity. (7)
    b) With an example write and explain the minimum cost spanning tree algorithm and its performance (7)

MODULE IV

17. a) Define Travelling Salesman Problem (TSP). Explain the basic steps that are to be followed to solve TSP using branch and bound. Illustrate with an example. (10)
    b) Discuss the elements of dynamic programming by considering the matrix chain multiplication problem. (4)

OR

18. a) Explain the concept of Backtracking. Explain how 4 Queen problem can be solved using backtracking. Draw the state space tree corresponding to 4 Queen problem. (7)
    b) Discuss Floyd-Warshall algorithm for all pair shortest path problem. With an example Solve the instance using the algorithm. (7)

MODULE V

19. a) Prove that CLIQUE problem is NP Complete. (7)
    b) Define approximation algorithm. Give an approximation algorithm for bin packing using first fit heuristic and give its approximation ratio. (7)

OR

20. a) Write randomized quicksort algorithm and perform its expected running time analysis. (7)
    b) Discuss the advantages of randomized algorithms over deterministic algorithms. Discuss Las Vegas and Monte Carlo algorithms with a suitable example. (7)

****