Reg. No. : | | | | | | | | | | | | |

Question Paper Code : 80097

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2019.

Fourth Semester

Computer Science and Engineering

CS 8451 — DESIGN AND ANALYSIS OF ALGORITHMS

(Common to Information Technology/Computer and Communication Engineering)

(Regulation 2017)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — ($10 \times 2 = 20$ marks)

1. How do you measure the efficiency of an algorithm?
2. Prove that the if $f(n) = O(g(n))$ and $g(n) = O(f(n))$, then, $f(n) = \theta\, g(n)$.
3. Write the brute force algorithm to string matching.
4. What is the time and space complexity of Merge sort?
5. State the principle of optimality.
6. What is the constraint for binary search tree insertion?
7. State the principle of duality.
8. Define the capacity constraint in the context of maximum flow problem.
9. Define NP completeness and NP hard.
10. State Hamiltonian Circuit problem.

PART B — (5 × 13 = 65 marks)

11. (a) (i) Solve the following recurrence equation :

(1) $T(n) = T(n/2) + 1$, where $n = 2^k$ for all $k \geq 0$ (4)

(2) $T(n) = T(n/3) + T(2n/3) + cn$, where '$c$' is a constant and '$n$' is the input size. (4)

(ii) Explain the steps involved in problem solving. (5)

Or

(b) (i) Write an algorithm for determining the uniqueness of an array. Determine the time complexity of your algorithm. (10)

(ii) Explain time-space trade off of the algorithm designed. (3)

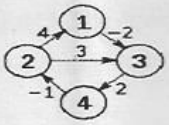
12. (a) What is the convex hull problem? Explain the brute force approach to solve convex-hull with an example. Derive the time complexity. (2 + 7 + 4)

Or

(b) Write the quicksort algorithm and explain it with an example. Derive the worst case and average case time complexity. (5 + 4 + 4)

13. (a) (i) Write the Floyd algorithm to find all pairs shortest path and derive its time complexity. (4 + 3)

(ii) Solve the following using Floyd's algorithm. (6)



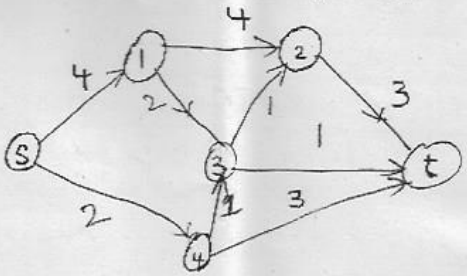
Or

(b) (i) Write the Huffman code algorithm and derive its time complexity. (5 + 2)

(ii) Generate the Huffman code for the following data comprising of alphabet and their frequency. (6)

$a : 1, b : 1, c : 2, d : 3, e : 5, f : 8, g : 13, h : 21$

14. (a) Determine the max-flow in the following network. (13)



Or

(b) Solve the following set of equations using Simplex algorithm : (13)

Maximize : $18x_1 + 12.5x_2$

Subject to : $x_1 + x_2 \leq 20$

$x_1 \leq 12$

$x_2 \leq 16$

$x_1, x_2 \geq 0$.

15. (a) Write an algorithm to solve the Travelling salesman problem and prove that it is a 2 time approximation algorithm. (13)

Or

(b) Write an algorithm for subset sum and explain with an example. (13)

PART C — ($1 \times 15 = 15$ marks)

16. (a) (i) Given a matrix of order $M \times N$, and two coordinates (p, q) and (r, s), which represents the top-left and bottom-right of a sub-matrix of the matrix, $M \times N$, calculate the sum of elements present in the sub-matrix in $O(1)$ time using dynamic programming. Determine the optimal sub-structure and write an algorithm. (10)

(ii) Prove that any algorithm that sorts by comparison, requires $\Omega(n \lg n)$ time. (5)

Or

(b) (i) The longest common subsequence (LCS) is the problem of finding the longest subsequence that is present in the given two sequences in the same order but not necessarily contiguously. Write an algorithm using dynamic programming that determines the LCS of two strings, 'x' and 'y' and returns the string 'z'. (10)

(ii) Prove that any algorithm that searches need to necessarily do $\Omega(\lg n)$ comparisons. (5)
