

**DESIGN AND ANALYSIS OF ALGORITHMS**  
(CSBS 2203)

Time Allotted: 2½ hrs

Full Marks: 60

*Figures out of the right margin indicate full marks.*

*Candidates are required to answer Group A and  
any 4 (four) from Group B to E, taking one from each group.*

*Candidates are required to give answer in their own words as far as practicable.*

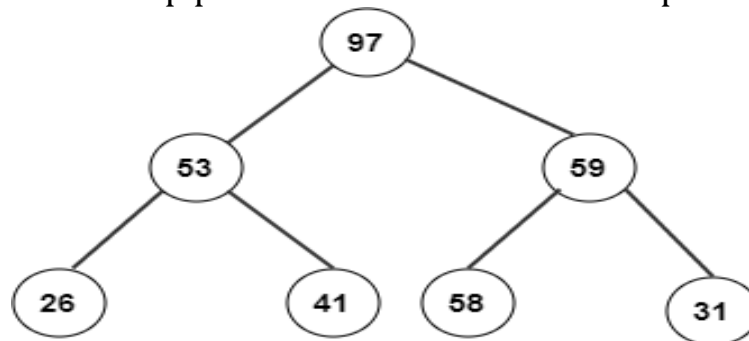
**Group – A**

1. Answer any twelve:

12 × 1 = 12

*Choose the correct alternative for the following*

- (i) What is the auxiliary space complexity of merge sort?  
(a)  $O(1)$                       (b)  $O(\log n)$                       (c)  $O(n)$                       (d)  $O(n \log n)$
- (ii) Depth First Search is equivalent to which of the traversal in the Binary Trees?  
(a) Pre-order Traversal                      (b) Post-order Traversal  
(c) Level-order Traversal                      (d) In-order Traversal
- (iii) In heap sort, after deleting the last minimum element, the array will contain elements in?  
(a) Increasing sorting order                      (b) Decreasing sorting order  
(c) Tree inorder                      (d) Tree preorder.
- (iv) Consider the following heap after build-heap phase. What will be its corresponding array?



- (a) 26,53,41,97,58,59,31                      (b) 26,31,41,53,58,59,97  
(c) 26,41,53,97,31,58,59                      (d) 97,53,59,26,41,58,31
- (v) What is the solution to the recurrence relation  $T(n) = 2T(n/2) + n$ , where  $T(1) = 1$ ?  
(a)  $O(n)$                       (b)  $O(n \log n)$                       (c)  $O(n^2)$                       (d)  $O(2^n)$ .
- (vi) Which of the following problems can be solved using Dynamic Programming?  
(a) Longest Common Subsequence                      (b) Fractional Knapsack Problem  
(c) Activity Selection Problem                      (d) Depth First Search.
- (vii) The N-Queens problem is a classic example of a problem that can be solved using backtracking. In this problem, how many queens must be placed on an  $N \times N$  chessboard such that no two queens attack each other?  
(a)  $N$                       (b)  $N-1$                       (c)  $2N-1$                       (d)  $2N$ .
- (viii) Which data structure is most suitable for implementing best first branch and bound strategy?  
(a) Stack                      (b) Queue                      (c) Priority queue                      (d) Linked list.
- (ix) Which of the following is NOT true about a spanning tree of a graph?  
(a) It must contain all vertices of the graph                      (b) It must be connected  
(c) It cannot contain cycles                      (d) It must be a minimum-weight subgraph.
- (x) Which of the following statements about NP-complete problems is true?  
(a) NP-complete problems can be solved in polynomial time  
(b) If a problem is NP-complete, it means it is not solvable  
(c) NP-complete problems are a subset of NP problems  
(d) NP-complete problems can only be solved by brute-force algorithms.

*Fill in the blanks with the correct word*

- (xi) The \_\_\_\_\_ algorithm is used to find the maximum flow in a flow network.
- (xii) The recurrence relation for the Fibonacci sequence  $F(n) = F(n-1) + F(n-2)$  with base cases  $F(0) = 0$  and  $F(1) = 1$ . The time complexity of the naïve recursive Fibonacci algorithm is \_\_\_\_\_.
- (xiii) If a graph has 6 vertices and 2 cycles involving 3 vertices then total number of spanning trees can be made is \_\_\_\_\_.
- (xiv) If a polynomial-time algorithm exists for any NP-complete problem, then  $P =$  \_\_\_\_\_.

(xv) The time complexity of the dynamic programming algorithm for matrix chain multiplication is \_\_\_\_\_.

### Group - B

2. (a) Calculate order of time complexity from the given piece of pseudo code using Frequency Count Method.

```
A(){
    n=2^(2^k);
    for(i=1;i<=n; i++){
        j=2;
        while(j<=n){
            j=j^2;
            printf("HIT");
        }
    }
}
```

(b) Prove that the time complexity of the function  $f(n) = \log n!$ , can't be expressed using the Big Theta notation ( $\theta$ ).

[[CO1](Analyse/IOCQ)]

(c) Solve the following recurrence relation using the Recursion-Tree Method:

[[CO1](Apply/IOCQ)]

$$T(n) = \begin{cases} 1 & n=0 \\ T(n-1) + \log n & n>0 \end{cases}$$

[[CO2](Apply/IOCQ)]

**3 + 4 + 5 = 12**

3. (a) State and explain the Master Theorem with example for Non-Decreasing Functions.

[[CO1](Understand/LOCQ)]

(b) Suppose  $P(n) = a_0 + a_1.n + a_2.n^2 + \dots + a_m.n^m$ , that is the degree of  $P(n)$  is  $m$ . Prove that  $P(n) = O(n^m)$ .

[[CO1](Analyse/IOCQ)]

(c) Consider the sequence 1, 3, 7, 15, 31, 63... This sequence satisfies the recurrence relation  $a_n = 3a_{n-1} - 2a_{n-2}$ . Find the Generating Function for the relation.

[[CO2](Apply/IOCQ)]

**2 + 4 + 6 = 12**

### Group - C

4. (a) Evaluate the Best case and Worst case time complexities of Quick Sort Algorithm with example.

[[CO1](Understand/LOCQ)]

(b) Consider the given graph  $G_1$ . Let the shortest distance between vertex  $P$  and vertex  $T$  be  $d$  and number of paths with distance  $d$  be  $l$ . Find out the value of the expression  $l^2 + 2ld + d^2$ .

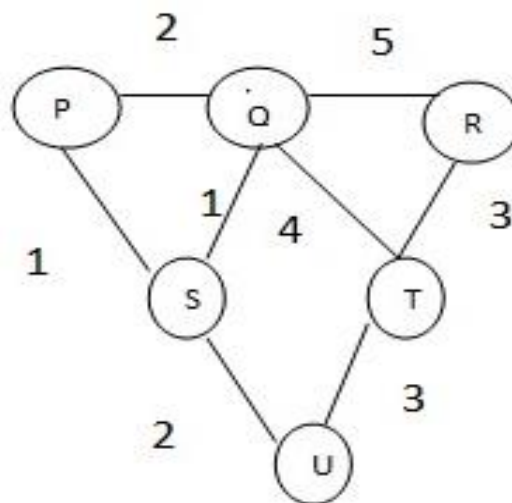


Fig. 1 Graph  $G_1$

(c) Describe briefly the Greedy method for solving optimization problems.

[[CO6](Evaluate/HOCQ)]

[[CO2](Remember/LOCQ)]

**(2 + 2) + 5 + 3 = 12**

5. (a) Perform Heap Sort algorithm on the given array  $L = \{81, 89, 9, 11, 14, 76, 54, 22\}$ . Illustrate each step to create the MAX-Heap and then removing the elements repeatedly to finally get the sorted array.

[[CO3](Apply/IOCQ)]

(b) Compare and contrast between Strassen's matrix multiplication method and standard matrix multiplication method using Divide & Conquer.

[[CO2](Understand/LOCQ)]

(c) Differentiate between BFS and DFS.

[[CO2](Understand/LOCQ)]

**5 + 5 + 2 = 12**

### Group - D

6. (a) Find out the optimal solution for the fractional Knapsack problem with capacity 60 is given below:

Weight = {5, 10, 20, 30, 40} & Profit = {30, 20, 100, 90, 160}.

[[CO2](Apply/IOCQ)]

(b) Find out the shortest path from vertex A to all remaining vertices for the following graph  $G_2$  using Dijkstra's Algorithm.

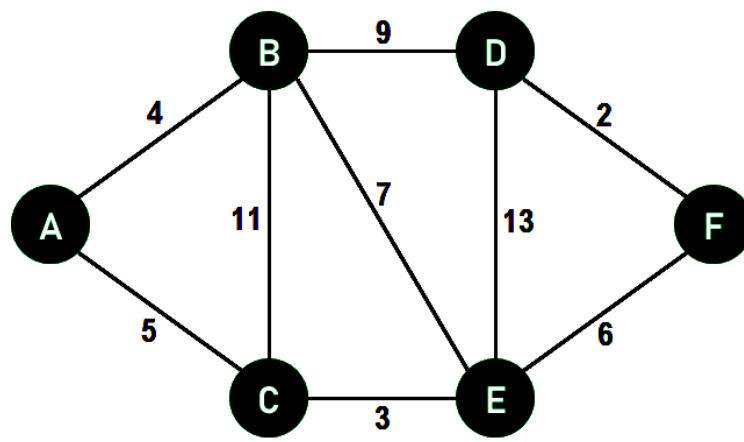


Fig. 2 Graph  $G_2$

(c) State Max -Flow and Min-Cut theorem.

[[CO2](Apply/IOCQ)]  
[[CO3](Remember/LOCQ)]  
**4 + 6 + 2 = 12**

7. (a) Given the jobs, their deadlines and associated profits as shown:  
 (i) Write the optimal schedule that gives maximum profit.  
 (ii) Are all the jobs completed in the optimal schedule?  
 (iii) What is the maximum earned profit?

Job	Deadline	Profit
J <sub>1</sub>	5	200
J <sub>2</sub>	3	180
J <sub>3</sub>	3	190
J <sub>4</sub>	2	300
J <sub>5</sub>	4	120
J <sub>6</sub>	2	100

(b) Find out Minimum number of scalar multiplication required to multiply the following chain of matrices:  $A_1$  ( $5 \times 15$ ),  $A_2$  ( $15 \times 10$ ),  $A_3$  ( $10 \times 5$ ),  $A_4$  ( $5 \times 25$ )

[[CO2](Apply/IOCQ)]  
[[CO4](Apply/IOCQ)]

(c) Consider the given directed weighted graph  $G_3$ . Using Floyd-Warshall Algorithm, find the shortest path distance between every pair of vertices.

[[CO3](Apply/IOCQ)]

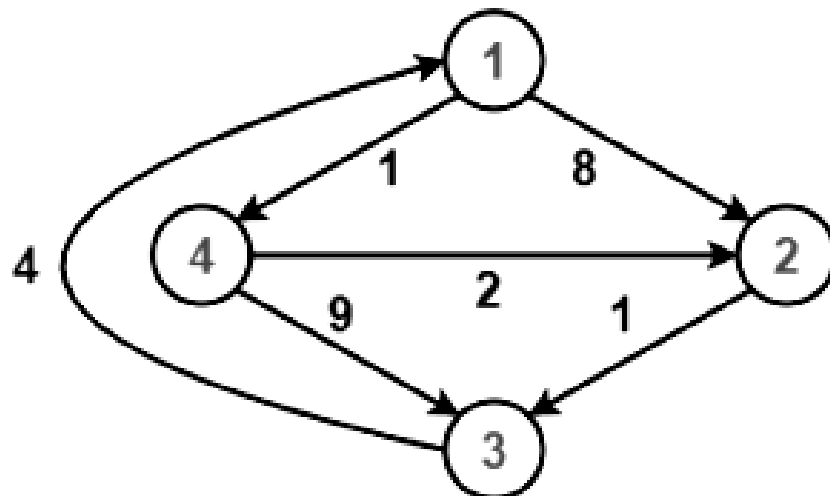


Fig. 3 Graph  $G_3$

**3 + 5 + 4 = 12**

**Group - E**

8. (a) Give a recursive algorithm MATRIX-CHAIN-MULTIPLY ( $A, s, i, j$ ) that actually performs the optimal matrix-chain multiplication, given the sequence of matrices  $\langle A_1, A_2, \dots, A_n \rangle$ , the  $s$  table computed by MATRIX-CHAIN-ORDER, and the indices  $i$  and  $j$ . (The initial call would be MATRIX-CHAIN-MULTIPLY ( $A, s, 1, n$ )).  
 (b) Explain with example the Ford-Fulkerson method to solve the maximum flow problem.  
 (c) Define classes P, NP and NP complete.

[[CO3](Understand/LOCQ)]  
[[CO3](Understand/LOCQ)]  
[[CO5](Remember/LOCQ)]  
**3 + 6 + 3 = 12**

9. (a) Define the basic concepts of backtracking with the help of neat flow diagram showing “Dead End” and “Success”. Hence, solve 4-Queens and 8-Queens problem using the above technique.  
 (b) Briefly discuss the concepts of Approximation Ratio and Polynomial Time in case of Approximation Algorithms.  
 (c) Explain the significance of SAT in NP-complete problem.

[[CO4](Apply/IOCQ)]  
[[CO5](Remember/LOCQ)]  
[[CO5](Remember/LOCQ)]  
**6 + 4 + 2 = 12**

Cognition Level	LOCQ	IOCQ	HOCQ
Percentage distribution	37.5	57.29	05.2

**Course Outcome (CO):**

- CO1.** Analyze the time and space complexity of an algorithm using asymptotic notations.
- CO2.** Distinguish the algorithms to solve optimization problems.
- CO3.** Illustrate the suitable algorithmic technique to find out a solution of decision making problems.
- CO4.** Apply the appropriate algorithmic techniques to solve the mathematical puzzles.
- CO5.** Describe different NP-hard and NP-complete problems and necessity of approximation algorithms.
- CO6.** Develop algorithms to solve common engineering problems.

*\*LOCQ: Lower Order Cognitive Question; IOCQ: Intermediate Order Cognitive Question; HOCQ: Higher Order Cognitive Question.*