*Basavarajeswari Group of Institutions*

# BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

(Autonomous Institute under Visvesvaraya Technological University, Belagavi)

| USN | | | | | | | | | | | Course Code | 2 | 1 | C | S | 4 | 3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

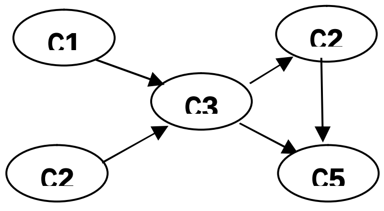
**Fourth Semester B.E. Degree Examination, Sept/Oct 2023**

## DESIGN AND ANALYSIS OF ALGORITHMS

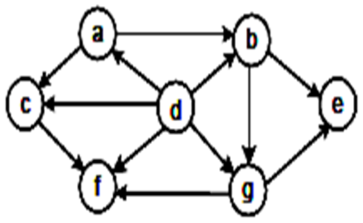
**(Common to CSE & AIML)**

**Time: 03 Hrs.** **Max. Marks: 100**

**Note**: Answer any **FIVE** full questions, choosing at least **ONE** question from each **MODULE**

| Q No | | Question | Marks | (RBTL:CO: PI) |
|---|---|---|---|---|
| | | **MODULE - I** | | |
| 1 | a | Define algorithm. With a neat flowchart explain the ***problem-solving technique***. | 10 | (2.1.1.3.1) |
| | b | What are the various basic ***asymptotic efficiency classes***? Explain ***Big Oh***, ***Big Omega,*** and ***Big Theta*** asymptotic notations | 10 | (2.1.1.3.1) |
| | | **OR** | | |
| 2 | a | Write the ***Tower of Hanoi*** algorithm and analyze the time complexity of given recursive algorithm | 07 | (3.2.1.3.1) |
| | b | Design an algorithm to search an element in an array using ***sequential search***. Discuss the Best-case worst case and average case efficiency of this algorithm | 07 | (3.2.1.3.1) |
| | c | Explain the general plan for analyzing the efficiency of a recursive algorithm. Write the algorithm to find a ***factorial*** of a given number. Derive its efficiency. | 06 | (2.1.1.3.1) |
| | | **MODULE - II** | | |
| 3 | a | Write bubble sort algorithm and analyze the time complexity of algorithm | 07 | (3.2.1.6.1) |
| | b | Write a ***binary search*** algorithm and derive its ***time complexity***. | 07 | (3.2.1.6.1) |
| | c | Apply topological sorting methods ***DFS-BASED and SOURCE REMOVAL*** for the following given graphs below and write the topological sequence.  | 06 | (3.2.1.3.1) |
| | | **OR** | | |
| 4 | a | Write selection sort algorithm and analyze the time complexity of algorithm | 07 | (3.2.1.6.1) |
| | b | Write an algorithm for ***insertion sort*** and derive its ***time complexity***. Apply the same to ***sort*** given list 74, 32, 93, 17, 73, 31, 44, 55, 20 | 07 | (3.2.1.6.1) |

c Apply topological sorting methods ***DFS-BASED and SOURCE REMOVAL*** for the following given graphs below and write the topological sequence. **06** **(3.2.1.3.1)**



## MODULE - III

**5** **a** Write an algorithm for ***quick sort*** and discuss worst case analysis of ***Quick Sort*** and derive time complexity with example. **07** **(3.3.1.6.1)**

**b** Apply ***Strassens's Matrix Multiplication*** algorithm to multiply the following matrices **07** **(3.3.1.3.1)**

$$A=\begin{bmatrix}4 & 5\\ 1 & 3\end{bmatrix} \qquad B=\begin{bmatrix}0 & 2\\ 1 & 3\end{bmatrix}$$

**c** Find the order of growth for the following recurrence relations **06** **(2.3.1.3.1)**
i. T(n) = 2 T (n/2) + O(n) ii. T(n) = 3 T (n/2) + O(n2).
iii. T(n) = 16T(n/2) + O(n) iv. T(n) = 2T(n/2) + nlogn

**OR**

**6** **a** Apply quicksort to ***sort*** the given list ***5, 3, 1, 9, 8, 2, 4, 7***in ascending order. Draw its ***recursive partition tree***. **07** **(3.3.1.3.1)**

**b** Write an algorithm to find ***maximum and minimum*** element in a given array using divide-and-conquer method. Derive its time complexity. **07** **(3.3.1.6.1)**

**c** Write an algorithm for ***merge sort*** and derive its ***time complexity***. **06** **(3.3.1.6.1)**

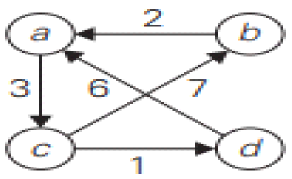
## MODULE - IV

**7** **a** Solve the given Knapsack Problem using dynamic programming approach and find the optimal solution. **12** **(3.4.1.3.1)**

| ITEM | WEIGHT | PROFIT |
|---|---|---|
| 1 | 2 | 12 |
| 2 | 1 | 10 |
| 3 | 3 | 20 |
| 4 | 2 | 15 |

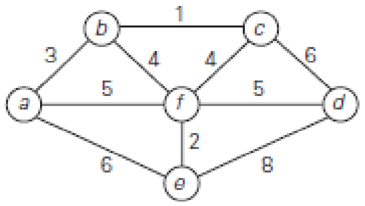
**Capacity W=5**

**b** Apply Floyd's Algorithm to find all pair shortest path for the given graph. **08** **(3.4.1.3.1)**



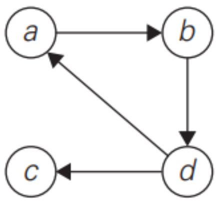
**OR**

**8** **a** Apply Prim's algorithm to the following graph. **08** **(3.4.1.3.1)**



**b** Write Kruskals algorithm and discuss with example. **06** **(2.4.1.6.1)**

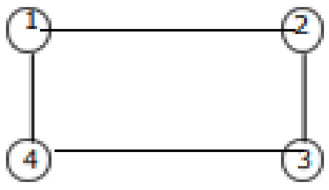
**C** Apply Warshals's Algorithm to find Transitive Closures for the given graph. **06** **(3.4.1.3.1)**



**MODULE - V**

**9** **a** Apply Backtracking Approach for **4-Queens Problem** to find all possible solutions with appropriate State Space tree structure. **10** **(3.5.1.6.1)**

**b** Apply Backtracking Approach to assign color to the given graph. M=3, {**RED, GREEN, BLUE}** **10** **(3.5.1.3.1)**



Gra p h

**OR**

**10** **a** Apply Backtracking Approach to solve sum of subset problem for the instance d**=30, S= {5,10,12,13,15,18}** give all possible solution with state space tree construction. **08** **(3.5.1.6.1)**

**b** Write a note on **12** **(2.5.1.3.1)**
1. P
2. NP
3. NP-Complete
4. NP-Hard problems