ADVANCED ALGORITHMS
(CSEN 5201)

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) The tightest lower bound on the number of comparisons, in the worst case, for comparison-based sorting of N elements is of the order of
(a) N (b) N^2 (c) N logN (d) None of these.

(ii) Merge procedure in Merge sort algorithm of n elements takes
(a) O (n logn) time (b) O (n) time
(c) O (1) time (d) None of the mentioned.

(iii) The worst-case running time of Randomized Quicksort when run on a table of n numbers is
(a) O(lg n) (b) O(n) (c) O(n lg n) (d) $O(n^2)$.

(iv) The necessary features to apply Dynamic Programming to solve a problem are
(a) Overlapping sub problems (b) Independent sub problems
(c) Optimal substructure (d) Both (a) & (c)

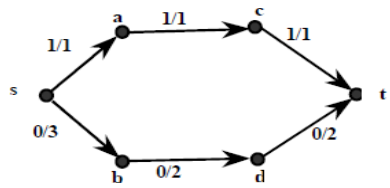
(v) Consider the following code snippet:

```
int fun(int n)
{
  int count = 0;
  for(int i = 0; i < n; i++)
      for(int j = i; j > 0; j--)
          count = count + 1;
  return count;
}
```

What is the time complexity of fun() in the worst case?
(a) θ (n) (b) θ (n^2) (c) θ (n logn) (d) θ (n loglogn)

(vi) Which of the following is the worst-case time complexity of BFS algorithm when the graph G = (V, E) is represented by adjacency list?
(a) O(|V|+|E|) (b) O(|V|^2) (c) O(|V|*|E|) (d) None of these.

(vii) What is the residual capacity of the augmenting path for the flow network shown below? Note that for each 'a/b' written beside it implies flow = 'a' and capacity = 'b'.



(a) 0.5 (b) 1 (c) 1.5 (d) 2.

(viii) Let K_n denote the complete graph on n nodes, in which every pair of nodes is connected by an edge. What is the size of the smallest vertex cover of K_n?
(a) n (b) n+1 (c) n/2 (d) n-1.

(ix) When an array A[1..n] containing n positive integers is converted into a max-heap, the leaves of nodes (viewed as a tree structure) are stored starting at index
(a) $\lfloor n/2 \rfloor$-1 (b) $\lfloor n/2 \rfloor$ (c) $\lfloor n/2 \rfloor$ +1 (d) $\lfloor n/2 \rfloor$ +2

(x) Which one of the following problems, in its commonly used formulation, does not currently have an algorithm that runs on a Deterministic Turing Machine in time that is polynomial in the size of the input?
(a) The Fractional Knapsack Problem
(b) The Travelling Salesman Problem
(c) The Single Source Shortest Path Problem
(d) The Minimum-Cost Spanning Tree Problem.

Fill in the blanks with the correct word

(xi) The space complexity of Merge-sort is ________ than that of Quicksort.

(xii) If you want to prove that a problem P is NP-Hard, then you need to show that, P can be reduced to any known NP-hard problem in polynomial time. The statement is _______ (True/ False).

(xiii) Strongly connected components of a directed graph can be found using ________.

(xiv) After an execution of partition subroutine of quicksort, at least one element of the array will occupy its final position as per the sorted ordering. This element is known as ______________________.

(xv) Dijkstra's algorithm for finding shortest path does not work if the graph has ____edge costs.

Group - B

2. (a) Asymptotically, how much time does insertion sort take when run on the following input: 2, 1, 4, 3, 6, 5, ..., n, n – 1? Give proper justification of your answer. *[CSEN 5201.1] [Understand/ LOCQ]*

(b) Is the array with values <16, 4, 10, 14, 7, 9, 3, 2, 8, 1> a max-heap? If yes give the reasons. If not, then HEAPIFY the above array to find the resultant array. Show all the steps. *[CSEN 5201.1] [Apply/IOCQ]*

(c) Is $2^{n+1} = O(2^n)$? Justify your answer. *[CSEN 5201.1] [Remember, Understand/ LOCQ]*

(1 + 3) + (1 + 5) + (1 + 1) = 12

3. (a) An array A contains n distinct positive integers. We want to find the k^{th} smallest integer in A (i.e., the k^{th} order statistic) for different values of k. Give an algorithm that will do this in O(n) expected time for every k, $1 \leq k \leq n$. Explain how it works. *[(CO3)(Analyse/LOCQ)]*

(b) An array A contains 2n+1 integers, n> 0, of which first element is 0, and out of the remaining 2n integers, k are positive (> 0) integers and (2n-k) are negative (< 0) integers. We want to re-arrange the numbers in the array using pairwise comparisons so that the first (2n-k) numbers are negative and the final k numbers are positive. Suggest a suitable method for achieving this. How many comparisons are needed by your method? *[(CO6)(Create/IOCQ)]*

8 + 4 = 12

Group - C

4. (a) G = (V,E) is a directed graph with 10 vertices numbered 1 through 10 and the following 15 directed edges: (1,2), (1,3), (1,4), (2,4), (3,4), (4,5), (4,6), (4,7), (5,7), (6,7), (7,8), (7,9), (7,10), (8,10), (9, 10). Each edge is directed from the first to the second vertex.
Using the DFS Algorithm, find the order in which vertices are traversed. List the discovery time and finishing time of vertices. Briefly outline the steps followed. How many different ways vertex 10 can be reached from 1? Explain. *[(CO2)(Solve/LOCQ)]*

(b) Find the topological sorting order of the above graph. Briefly explain the logic of your method. *[(CO3)(Apply,Evaluate/IOCQ)]*

8 + 4 = 12

5. (a) Let G = (V,E) be a weighted directed graph with 5 vertices and 8 edges. The vertices are numbered 1 through 5, and the edges are as follows: (1,2,8), (1,4,5), (1,5,3), (2,3,3), (2,5,-7), (4,3,2), (3,5,6), (4,5,5). Here (u,v,w) represents an edge from vertex u to vertex v of weight w.
Apply Dijkstra's algorithm to find the least weight path from the start node 1 to every other node. Show the steps and list the path weights obtained for every node. If you apply Bellman-Ford algorithm, justify whether you get the same path weights for every node. Compare the time-complexity of these algorithms. *[(CO2,CO4,CO5)](Apply,Evaluate/LOCQ)]*

(b) Assume edges are undirected in the graph given above. Edges and edge weights remain same. Which algorithm Prim's or Kruskal's (or none) can be applied to find the minimum-cost (weight) spanning tree in this graph with negative edge weight? Explain your answer. *[(CO2)(Apply,Analyse/IOCQ)]*

8 + 4 = 12

Group - D

6. (a) Matrices A, B and C, of dimensions 10 × 24, 24 × 6 and 6 × 17 respectively, are provided. You are told to ensure that when computing the matrix product AxBxC, the number of scalar multiplications will be a minimum. How would you parenthesize the product, and what would be the minimum number of scalar multiplications you would need to perform? *[(CSEN 5201.4)(Apply/IOCQ)]*

(b) You are taught in the class two single source shortest path algorithms. We can surely apply those algorithms to find the shortest paths between every pair of vertices in the graph by changing the source each time. Then why do we need to study a different "All-pairs shortest path" algorithm? Justify your answer briefly. *[(CSEN5201.5)(Analyse/IOCQ)]*

(c) Briefly explain Floyd – Warshall algorithm. *[CSEN 5201.2][Understand/LOCQ]*

(2 + 2) + 3 + 5 = 12

7. (a) A 0-1 Knapsack Problem has a knapsack of weight limit 12, and 6 items with the values and weights given below:

Item	1	2	3	4	5	6
Value	8	9	7	5	6	4
Weight	6	7	5	3	4	4

Solve the problem using dynamic programming and obtain the maximum total value that can be achieved. How will the solution change if it is specified that item 6 must be included in the knapsack? *[(CO3)(Apply/HOCQ)]*

(b) Solve the Fractional Knapsack Problem with the data given in (a). Show the change in the solution if it is specified that item 6 must be included entire (i.e., fully, not partially) in the knapsack. *[(CO2)(Analyze/LOCQ)]*

8 + 4 = 12

Group – E

8. (a) Differentiate clearly between the sets **P**, **NP** and **NPC**. Give an example of a problem in **NP** that is not known to be in **P** at the current time. *[(CO2)(Understand/HOCQ)]*

(b) Write an approximation algorithm to solve the VERTEX COVER problem. How close it is to the optimal solution? *[(CO6)(Design/IOCQ)]*

6 + 6 = 12

9. (a) State which problems are solvable in polynomial time and which are NP-hard?
(i) Hamiltonian Cycle
(ii) Subset Sum Problem
(iii) Set Cover Problem
(iv) Edge Cover Problem. *[CSEN 5201.2][Remember,Understand/LOCQ]*

(b) Prove that the Clique Decision problem is NP-hard by using the result that the 3-CNF-SAT problem is computationally hard. *[CSEN 5201.3][Apply/IOCQ]*

4 + 8 = 12

Cognition Level	LOCQ	IOCQ	HOCQ
Percentage distribution	44.79	40.63	14.58

Course Outcome (CO):

After the completion of the course students will be able to
CSEN5201.1. Remember time complexities of various existing algorithms in different situations
CSEN5201.2. Understand the basic principles of different paradigms of designing algorithms
CSEN5201.3. Apply mathematical principles to solve various problems
CSEN5201.4. Analyze the complexities of various algorithms
CSEN5201.5. Evaluate the performance of various algorithms in best case, worst case and average case
CSEN5201.6. Create/ Design a good algorithm for a new problem given to him/ her.

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