DESIGN AND ANALYSIS OF ALGORITHMS
(CSBS 2203)

Time Allotted : 3 hrs **Full Marks : 70**

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and <u>any 5 (five)</u> from Group B to E, taking <u>at least one</u> from each group.

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

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternative for the following: **10 × 1 = 10**

(i) An algorithm takes T(n) = log n! steps. What is the tightest bound for T(n) in big-Oh notation?
(a) O(n log log n) (b) O(n log n) (c) O(n(log n) n) (d) O(n log n).

(ii) Consider the following code snippet:
```
void fun(int n)
{
int p = 0;
for(int i = 1; i < n; i=i*2)
  p++;
    for(int j =1; j < p; j=j*2)
      print("%d%d", i, j);
return;
}
```
The time complexity of fun() in the worst case is
(a) O (n) (b) O (n^2) (c) O (n log n) (d) O (log log n)

(iii) How would you classify the time complexity of an algorithm for the 0/1 Knapsack problem with *n* objects and capacity *W* which runs in O(nW) time? Assume that the inputs to the problem are the sizes and profits of the ***n*** objects and the capacity W.
(a) polynomial (b) exponential (c) pseudo-polynomial (d) quadratic.

(iv) Worst case time complexity for inserting an element in a sorted array so that it stays sorted is
(a) O(1) (b) O(n) (c) O(n^2) (d) O(n log n).

(v) Consider a complete graph G with *four* vertices. Then the number of spanning tree of graph G is
(a) 15 (b) 8 (c) 16 (d) 13.

(vi) An algorithm is made up of two modules module1 and module2. If order of module1 is f(n) and module2 is g(n), then the order of the algorithm is
(a) max(f(n), g(n)) (b) min(f(n), g(n)) (c) f(n) + g(n) (d) f(n) * g(n)

(vii) The travelling salesman problem can be solved in
(a) polynomial time using dynamic programming algorithm
(b) polynomial time using branch-and-bound algorithm
(c) exponential time using dynamic programming algorithm or branch and bound algorithm
(d) polynomial time using backtracking algorithm.

(viii) f(n)=Θ(g(n)) if and only if g(n)= Θ(f(n)) denotes _______ property.
(a) transitivity (b) symmetry (c) transpose symmetry (d) reflexivity

(ix) Which one is true for the following?
(a) All NP-Hard problems are NP-Complete
(b) All NP-Complete problems are NP-Hard
(c) Some NP-Complete problems are NP-Hard
(d) None of these.

(x) Backtracking algorithm is implemented by constructing a tree of choices called as
(a) State-space tree (b) State-chart tree (c) Node tree (d) Backtracking tree.

Group - B

2. (a) With proper justification arrange the following functions in increasing order of growth 2^n $n^{3/2}$ nlogn $n^{\log n}$. [(CO1)(Analyse/IOCQ)]

(b) Let T(n) be the running time of the algorithm. The recurrence relation of T(n) is given below:

$$T(n) = \begin{cases} 1 & if\ n = 0 \\ 2T(n-1) + 1 & if\ n > 0 \end{cases}$$

Show that T(n) = Θ (2^n) using
(i) Recursion tree
(ii) Substitution method. [(CO1)(Apply/IOCQ)]

(c) Prove that (i) if f(n) = O(g(n)) then g(n) = Ω(f(n))
(ii) f(n) = Θ(g(n)) iff g(n) = Θ(f(n)). [(CO1)(Apply/IOCQ)]

4 + 4 + 4 = 12

3. (a) Write a divide–and–conquer quick sort algorithm and derive the best case and worst case time complexity of this algorithm. [(CO2)(Analyse/IOCQ)]

(b) Write an algorithm/function to implement priority queue using heap. [(CO4)(Apply/IOCQ)]

6 + 6 = 12

Group - C

4. (a) Write an algorithm for finding the minimal spanning tree of a graph, Also discuss its time complexity. [(CO2)(Analyse/IOCQ)]

(b) Given the weight vector (2, 3, 5, 7, 1, 4, 1) and the profit vector (10, 5, 15, 7, 6, 18, 3) and a Knapsack of capacity 15, find an optimal solution for the fractional knapsack problem. [(CO2)(Apply/IOCQ)]

8 + 4 = 12

5. (a) What is a fractional knapsack problem? [(CO2)(Understand/LOCQ)]

(b) Find a minimum spanning tree of the graph given bellow (Fig. 1) using Prim's algorithm. [(CO2)(Apply/IOCQ)]

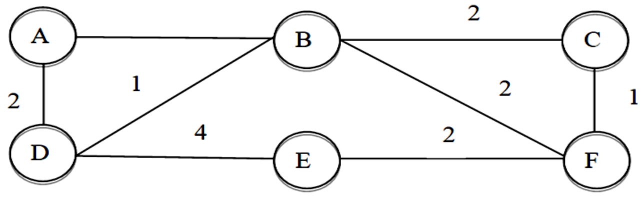

Fig. 1

(c) A set of seven jobs is given, with associated profit {3, 5, 20, 18, 1, 6, 30} and deadline {1, 3, 4, 3, 2, 1, 2} respectively, where the profit is earned if and only if a job is completed by its deadline. Only one machine is available for processing jobs and each job requires *one unit* of time on the machine to be completed. Find a sequence of jobs to be processed so that, the profit is maximized.

[(CO2)(Apply/IOCQ)]

2 + 5 + 5 = 12

Group - D

6. (a) Apply the Ford Fulkerson method to solve the following maximum flow problem (Fig. 2), where for each edge, the first value represents the flow value and the second stands for the capacity of the edge. Consider vertex S as source and vertex T as sink. [(CO3)(Apply/IOCQ)]

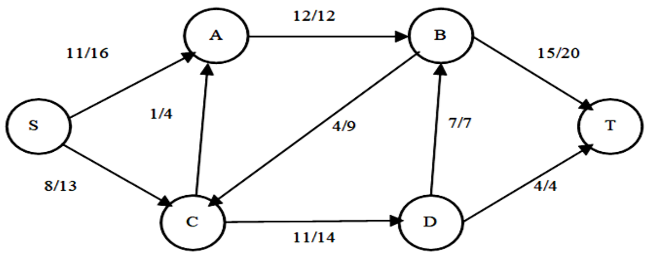

Fig. 2

(b) Using the given edge cost matrix for the vertex set {1, 2, 3, 4}, find a dynamic programming solution of the travelling salesperson problem, where the origin is vertex 1. [(CO2)(Apply/IOCQ)]

	1	2	3	4
1	0	5	6	8
2	10	0	13	8
3	15	9	0	9
4	20	10	12	0

6 + 6 = 12

7. (a) Differentiate between divide-and-conquer and dynamic programming.

[(CO2)(Analyse/IOCQ)]

(b) Use Bellman-Ford's algorithm to solve the single-source shortest path problem in the below graph (Fig. 3), using vertex ***B*** as source. Show all the passes and define the time complexity of the algorithm. [(CO2)(Apply/IOCQ)]

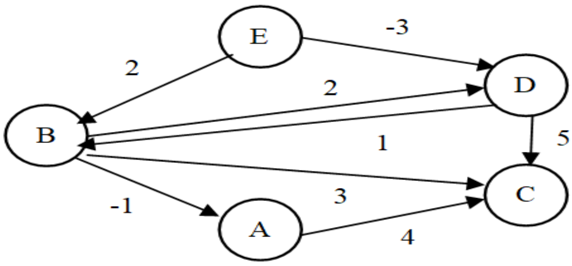

Fig. 3

4 + (6 + 2) = 12

Group - E

8. (a) Give an example of NP-hard decision problem which is not NP-complete. Give reasons for the same. [(CO5)(Analyse/IOCQ)]

   (b) Resolve the 8 queens problem in terms of branch and bound strategy. Compare this approach with backtracking. [(CO2)(Analyse/IOCQ)]

5 + (4 + 3) = 12

9. (a) Write the least-cost search method to find the solution of the 15 puzzle problem. Will this method guarantee solution for all instances of the problem? [(CO6)(Apply/IOCQ)]

   (b) Prove that clique decision problem is NP-complete. [(CO5)(Analyse/IOCQ)]

8 + 4 = 12

Cognition Level	LOCQ	IOCQ	HOCQ
Percentage distribution	2.08	97.92	0

Course Outcome (CO):

After the completion of the course students will be able to

1. Demonstrate how the time complexity of an algorithm is defined and analyze the asymptotic performance of algorithms.
2. Understand basic algorithm designing techniques such as divide and conquer, greedy, dynamic programming, branch and bound, backtracking and analyze them.
3. Explain the graph algorithms such as BFS, DFS, Ford Fulkerson Method, etc and analyze them.
4. Synthesize efficient algorithms in common engineering design situations.
5. Exploration of various research problems in algorithm like NP-hard and NP-complete problems.
6. Explain what an approximation algorithm is, and the benefit of using approximation algorithms.

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