No. of Printed Pages—3

**648 (DAA)**

**[ 18–HRGPKAC–MCA5S ]**

**MCA 5th Semester Exam., 2018**

# DESIGN AND ANALYSIS OF ALGORITHM

*Full Marks* : 80

*Time* : 3 hours

*The figures in the margin indicate full marks for the questions*

**1.** Answer any *five* from the following questions : 2×5=10

*(a)* Define time complexity and space complexity.

*(b)* What are O (big oh) and  (omega) notations?

*(c)* Give the recurrence relation for the worst case behaviour of merge sort.

*(d)* Write the drawback of Greedy algorithm.

*(e)* What does dynamic programming have in common with divide and conquer?

*(f)* Define live node and dead node.

**2.** Answer any *three* from the following questions : 4×3=12

*(a)* Show the steps when the numbers 123, 23, 1, 43, 54, 36, 75, 34 are sorted using merge sort.

*(b)* Explain how greedy method can be applied to solve the knapsack problem.

*(c)* State Cook's theorem. What is satisfiability problem?

*(d)* Define in-place and stable sorting algorithms.

**3.** Answer any *three* from the following questions : 6×3=18

*(a)* Solve the following instance of the 0/1, knapsack problem given in the knapsack capacity as $W$ 5 :

| *Items* | *Weight* | *Value* |
|---|---|---|
| 1 | 2 | 12 |
| 2 | 1 | 10 |
| 3 | 3 | 20 |
| 4 | 2 | 15 |

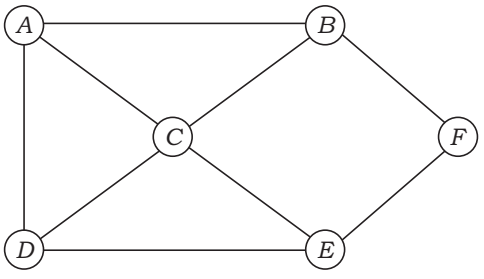
*(b)* Define *n*-Queen's problem. Give the solution for 4-Queen's problem. 2+4=6

*(c)* Write an algorithm for linear search.

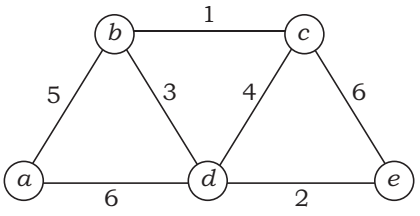
*(d)* Define bounding function. Differentiate between explicit constraints and implicit constraints.

**4.** Answer any *four* from the following questions : 10×4=40

*(a)* Write an algorithm to determine Hamiltonian cycle in a given graph using backtracking. For the following graph, determine the Hamiltonian cycle :

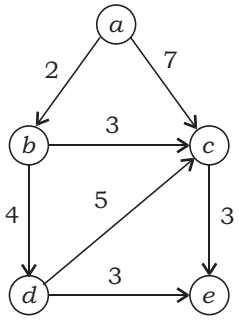



*(b)* Write and apply Kruskal's algorithm to find a minimum spanning tree for the following graph :



*(c)* Describe the branch and bound technique. Differentiate between FIFO and LIFO branch and bound search. 6+4=10

*(d)* Write and apply Dijkstra's algorithm to find a shortest path for the following graph :




*(e)* Discuss the relationship among *P*, *NP*, *NP*-hard and *NP*-complete classes.

★ ★ ★