Madras University

M.C.A DAA Question paper

Time: Three hours **Maximum: 75 marks**

PART A - [5 x 5 = Marks 25]

Answer ALL questions.
Each question carries 5 marks.

1. (a) Show that if ¦1 (n) = 0 [¦2 (n)], then ¦2 (n) = 0 [¦1 (n)].

Or

(b) Explain the meaning of deterministic algorithms and infinite algorithms.

2. (a) Jot down four vertices A,B,C and D and construct all possible spanning trees.

Or

(b) Show that if a tree has n vertices, then it has (n-1) edges.

3. (a) With reference to the travelling salesman problem, is the cost matrix always symmetric? Justify your answer.

Or

(b) What is an Eulerian Walk? Under What conditions, does a multigraph have an Eulerian walk?

4. (a) Why are backtracking algorithms called method of last resort?

Or

(b) Give an efficient algorithm to determine whether or not a graph can be painted with just two colours.

5. (a) Determine the probability that a randomly selected tour in an n city travelling salesman problesm is the optimal one.

Or

(b) With reference to the knapsack problem, state the condition under which it is clearly optimal to pack all the objects in the knapsack.

PART B - [5 x 10 = Marks 50]

Answer any FIVE questions.
Each question carries 10 marks.

6. Determine the product of 5678 and 349 using divide and conquer technique.

7. With the help of a suitable example, explain Kruskal's algorithm for determining the minimal spanning tree.

8. Design a dynamic programming algorithm to find an optimal binary search tree for a set of keys with given probabilities of access.

9. How many solutions are there to the eight queens problem? How many distinct solutions are there if we do not distinguish solutions that can be transferred into one another by rotations and reflections?

10. Prove or give a counter example : If a graph is bicoherent, then it is biconnected.

11. Arrange the following numbers in ascending order using three way merge-sort: 26,15,47,05,99,47,90,50,70,47,98,26.

12. State the knapsack problem and explain how it can be solved using a branch and bound algorithm.

13. Explain how you will determine the total number of partitions of a positive integer using a recursive algorithm.