Total No. of Questions : 12] [Total No. of Pages : 3

P595

[4064] - 585

B.E. (Computer Engineering)

DESIGN & ANALYSIS OF ALGORITHMS

(Sem. - I) (2008 Course) (410441)

Time : 3 Hours] *[Max. Marks :100*

Instructions to the candidates:

1) ***Answer any three questions from each section.***
2) ***Answers to the two sections should be written in separate books.***
3) ***Figures to the right indicate full marks.***
4) ***Assume suitable data, if necessary.***

SECTION - I

Q1) a) Apply the principles of mathematical induction to prove that the algorithm for function SQUARE, given below, is correct. **[3]**

```
Function SQUARE (n)
if (n = 0) then return O
else return 2 n + SQUARE (n–1) – 1
```

b) Derive the time complexity of Quick sort algorithm. **[5]**

c) Write Dijkstra's algorithm for a directed graph. Compare it with all-pairs shortest path algorithm. **[8]**

d) Compare "Divide and conquer" with "Greedy" method. **[2]**

OR

Q2) a) Describe the recurrence relation for computing time of Merge sort and solve it. **[5]**

b) What are NP-Hard and NP-complete problems? **[2]**

c) Solve the following instance of "job sequencing with deadlines" problem: n = 7, **[3]**

profits $(p_1, p_2, p_3, \ldots\ldots\ldots, p_7) = (3, 5, 20, 18, 1, 6, 30)$ and deadlines $(d_1, d_2, \ldots\ldots, d_7) = (1, 3, 4, 3, 2, 1, 2)$

d) Design and analyze an algorithm to solve the exponentiation problem using divide and conquer strategy. **[8]**

P.T.O.

Q3) a) What is the principle of optimality? Give example of a problem where principle of optimality does not hold. **[2]**

b) Solve the instance of 0/1 knapsack problem using dynamic programming: n = 4, m = 25 **[8]**

$(p_1, p_2, p_3, p_4) = (10, 12, 14, 16)$,

$(w_1, w_2, w_3, w_4) = (9, 8, 12, 14)$

c) Explain how dynamic programming is used to obtain optimal solution for traveling salesperson problem. Also explain why this technique is not used to solve TSP for large number of cities. **[6]**

OR

Q4) a) What are the various ways of implementing the knapsack problem? Explain. **[3]**

b) Write a algorithm for solving the problem of optimal binary search tree. **[10]**

c) Explain why flow shop scheduling problem is not solved using dynamic programming strategy for more than 2 tasks per job. **[3]**

Q5) a) Write an algorithm for N Queens problem which uses backtracking. Explain the implicit and explicit constraints of this problem. What is the time complexity of your algorithm? **[8]**

b) Solve the traveling sales person problem instance, by using branch and bound method, defined by the cost matrix given below. **[8]**

$$\begin{bmatrix} \alpha & 11 & 10 & 9 & 6 \\ 8 & \alpha & 7 & 3 & 4 \\ 8 & 4 & \alpha & 4 & 8 \\ 11 & 10 & 5 & \alpha & 5 \\ 6 & 9 & 5 & 5 & \alpha \end{bmatrix}$$

OR

Q6) a) Explain the problem of m-colourability for planar and non-planar graphs, and write an algorithm for m-colourability problem for general graph(non-planar) using backtracking strategy. **[8]**

b) Explain in detail how 0/1 knapsack problem can be solved using least cost branch and bound strategy. **[8]**

SECTION - II

Q7) a) Show that a polynomial time algorithm that makes polynomial number of calls to a polynomial time subroutine may run in exponential time. **[4]**

b) Prove that the clique decision problem (CDP) is NP - complete. **[8]**

c) Explain how Directed Hamiltonian Cycle (DHC) reduces to Traveling sales person decision problem. **[6]**

OR

Q8) a) What are non-deterministic algorithms? Give example. Explain where they are used while classifying algorithms. **[5]**

b) Prove that the chromatic number decision problem (CNDP) is NP-hard. **[8]**

c) Prove that partition reduces to minimum finish time preemptive job scheduling problem (m > 1). **[5]**

Q9) a) Differenciate between the models of computation for parallel computation. **[4]**

b) When are parallel algorithms "work. optimal". **[2]**

c) Describe a parallel algorithm for finding the connected components of a given graph. **[10]**

OR

Q10) a) Explain pointed doubling problem and why it can be solved parallely. **[5]**

b) Explain the odd-even Merge technique for efficient parallelization of merging. **[5]**

c) Prove that n arbitrary keys can be sorted in $O(\log^2 n)$ time using n EREW PRAM processors. **[6]**

Q11) a) Comment on the sequential and parallel technique for solving convex hull problem. **[8]**

b) Which are the useful algorithmic strategies for image edge defection problem? Explain why and also mention how underlying data structure affects the choice of algorithm. **[8]**

OR

Q12) a) Explain a deadlock detection and avoidance algorithm. Discuss the algorithmic strategy used. **[8]**

b) Evaluate all options for implementing Huffman's problem. **[8]**

❄❄❄