2018

BCA 3rd Semester Examination

DESIGN AND ANALYSIS OF ALGORITHMS

PAPER—2101

Full Marks : 70

Time : 3 Hours

The figures in the right-hand margin indicate full marks.

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

Illustrate the answers wherever necessary.

Answer Q. No. *1* and any *four* from the rest.

1. Answer any *five* questions : 5×2

(a) Which sorting technique has worst-case time complexity O (n^2) but average-case time complexity O ($n.\log_2^n$)?

(b) What do you mean by 'P' complexity class ?

(c) Write the different properties of an algorithm.

(d) What is "greedy-choice property" ?

(e) Define Big-Theta (θ) notation of complexity.

(f) What is asymptotic notation in algorithm ?

(g) What is a recursive algorithm ?

(h) Why the binary search is called so ?

2. (a) Explain the Merge sort technique with example.

(b) Show that average-case time complexity of quick sort is $O(n\log_2^n)$.

(c) Write the recursive solution of Tower of Hanoi Problem.

7+5+3

3. (a) What do you understand by "dynamic programming" of algorithm design Paradigm ?

(b) Describe Dijkstra's algorithm to find single-source shortest path of a graph, with example.

(c) What's the Travelling Salesman Problem ?

4+7+4

4. (a) Describe Breadth-First Search of graph traversal algorithm with example.

   (b) Write an algorithm to find UNION of two disjoint sets.

   (c) Explain graph coloring problem. 6+5+4

5. (a) Write binary search algorithm using Divide-and-conquer approach.

   (b) What is circuit satisfiability problem ?

   (c) What's the backtracking method of algorithm design ? Using backtracking algorithm give a solution to 8-queen problem. 6+4+(2+3)

6. (a) Explain class P and class NP problems in detail. What is an NP Complete problem ?

   (b) Write down Kruskal's algorithm for finding minimum spanning tree of a graph.

   (c) Explain approximation algorithm in detail.

   (6+2)+4+3

7. (a) Write down the 0-1 Knapsack problem algorithm using dynamic programming strategy. What is the time complexity of this algorithm ?

   (b) Explain how a greedy algorithm works with an example.

   (c) Explain tail recursion and the advantage of using tail recursive function over non-tail recursive function.

(6+2)+3+4

———