BIRLA INSTITUTE OF TECHNOLOGY, MESRA, RANCHI
(END SEMESTER EXAMINATION)

CLASS: MCA **SEMESTER : III**
BRANCH: MCA **SESSION : MO/19**

SUBJECT: CA503 COMPUTER ALGORITHM DESIGN

TIME: 3 HOURS **FULL MARKS: 50**

INSTRUCTIONS:
1. The question paper contains 5 questions each of 10 marks and total 50 marks.
2. Attempt all questions.
3. The missing data, if any, may be assumed suitably.
***4*. Before attempting the question paper, be sure that you have got the correct question paper.**
5. Tables/Data hand book/Graph paper etc. to be supplied to the candidates in the examination hall.

Q.1(a) Prove that every polynomial of degree *k*, $P(n) = a_k n^k + a_{k-1} n^{k-1} + \ldots a_0$, with $a_k > 0$ belongs to $\Theta(n^k)$. [5]

Q.1(b) There exist algorithms for matrix multiplication with a better asymptotic efficiency than the cubic complexity of the definition-based algorithm. Because of their much larger multiplicative constants, however, the value of these more sophisticated algorithms is mostly theoretical. Discuss the above statement in view of *theoretical versus empirical analysis* of algorithms. [5]

Q.2(a) Solve the following recurrence exactly for *n* a power of 2. [5]
$T(n) = 2T(n/2) + \lg n$, for all $n > 1$
$T(1) = 1$.
Express your answer as simply as possible using the Θ notation.

Q.2(b) Discuss Dijkstra's algorithm for *single source shortest paths* problem. Does Dijkstra's algorithm work for graphs with negative weights? Justify your answer. [5]

Q.3(a) Apply quicksort to sort the list M, E, R, G. E, S, O, R, T in alphabetical order. Find the element whose position is unchanged in the sorted list. [5]

Q.3(b) If only the number of additions and the number of multiplications are taken into account, which one of the following algorithm would be perfect and why? – Strassen's algorithm – brute-force algorithm. [5]

Q.4(a) Using dynamic programming approach develop an algorithm to solve the 0/1 Knapsack problem. Compute time complexity of your approach. [5]

Q.4(b) Design a dynamic programming algorithm for the 'change-making-problem'. Given an amount *n* and unlimited quantities of coins of each of the denominations $d_1, d_2, \ldots, d_m$, find the smallest number of coins that add up to *n* or indicate that the problem does not have a solution. [5]

Q.5(a) Explain how one can identify connected components of a graph by using depth-first-search algorithm. [5]

Q.5(b) What do you mean by a randomized algorithm? Discuss its various classes with suitable example(s). [5]

:::::27/11/2019:::::M