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

CLASS: MCA
BRANCH: MCA

SEMESTER : III
SESSION : MO/14

SUBJECT: MCA3005 FUNDAMENTALS OF COMPUTER ALGORITHMS

TIME: 3.00 HOURS

FULL MARKS: 60

INSTRUCTIONS:
1. The question paper contains 7 questions each of 12 marks and total 84 marks.
2. Candidates may attempt any 5 questions maximum of 60 marks.
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) What do you mean by time efficiency of an algorithm? How do you express the performance of an algorithm? [6+6]

Q.1(b) Design a linear time algorithm for finding the first non-repeating character in a text (over English Alphabet).

Q.2(a) If $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then $f(n) = \Omega(h(n))$. [4+8]

Q.2(b) Solve the following recurrence relations:
i) $T(n) = T(n/2) + T(n/2) + n$, and $T(1) = 1$.
ii) $T(n) = 3\,T(n^{1/3}) + \log_3 n$, and $T(3) = 1$.

Q.3(a) Discuss the basic idea behind *Greedy* approach. [4+8]

Q.3(b) Discuss PRIM's algorithm and compare between PRIM's and KRUSKAL's algorithms. Does both the approaches follow the Greedy approach? Justify your answer. Prove or disprove: suppose that each edge weight of a connected graph is decreased by 1 (one) after finding an MST. Now, the tree generated earlier is unaffected to result MST.

Q.4(a) Consider the following problem. The input is a set S containing n real numbers and one real number x. Design an algorithm to determine whether there are two elements of S whose sum is exactly x. The algorithm should run in time O (n log n). [6+6]

Q.4(b) Explain *Strassen's* matrix multiplication. Analyze the time complexity of the approach.

Q.5(a) Discuss the features of *Dynamic programming*. What are the drawbacks of this approach? [5+7]

Q.5(b) Explain in details how Dynamic programming can be applied to solve *Knapsack problem*.

Q.6(a) State the principle of Backtracking. [4+8]

Q.6(b) Explain the *sum of subset* problem and its algorithmic implementation.

Q.7(a) What is non-deterministic algorithm? Differentiate between *Monte Carlo* and *Las Vegas* algorithms. Explain the importance of Randomized algorithm. [6+6]

Q.7(b) Suppose we need to find the i^{th} smallest element from an unsorted list of n elements (assume that $n > i$). Design a randomized algorithm in this purpose. Also, analyze the running time of the designed algorithm.

::::: 28112014 E :::::