M. TECH. –II (COMPUTER ENGINEERING) (CBCS – 2015 COURSE) : SUMMER - 2018

SUBJECT: ADVANCED COMPUTER ALGORITHMS

Day: **Wednesday** | **S-2018-2999** | Time: **11.00 AM TO 02.00 PM**
Date: **13/06/2018** | Max Marks: 60

N.B.:

1) All questions are **COMPULSORY.**
2) Figures to the right indicate **FULL** marks.
3) Answers to both the sections should be written in **SEPARATE** answer book.
4) Assume suitable data, if necessary.

SECTION-I

Q.1 Give RAM, RASP and Pidgin ALGOL programs to do the following : **(10)**

1. Compute n! Given input n
2. Read n positive integers followed by an endmarker (0) and then print the n numbers in sorted order
3. Accept all inputs of the form $1^n2^{n^2}0$

OR

What happens to the computing power of RAM or RASP if both MULT and ADD are removed from the instruction repertoire? How is cost of computation affected? **(10)**

Q.2 Write an algorithm to reverse the order of items in a list? Prove that your algorithm works correctly and state its time complexity. **(10)**

OR

Compare the sorting methods 'Merge sort and Quick sort' algorithms. Device data sets that compare both the average and worst case times for these two algorithms **(10)**

Q.3 What is sorting problem? Describe 'sorting by comparison'. How is expected time of Heap sort equal to O(nlogn)? **(10)**

OR

Write the algorithm for recursive binary search and solve its recurrences relations to obtain its expected running time. **(10)**

SECTION-II

Q.4 By considering the complete graph with n vertices, Show that the number of spanning trees in an n vertex graph can be greater than $(2^{n-1}-2)$ **(10)**

OR

Describe in detail Strassen's algorithm. Use the algorithm to compute the product: **(10)**

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} \quad \begin{bmatrix}5 & 6\\7 & 8\end{bmatrix}$$

Q.5 Write LC branch and bound algorithm for the knapsack problem. **(10)**

OR

What is Hamiltonian cycles? Determine the order of magnitude of the worst case computing time for the backtracking procedure that finds all Hamiltonian cycles **(10)**

Q.6 Write a polynomial algorithm to test for 2-Satisfiablity **(10)**

OR

Explain the need for approximation algorithms and how they can be used for NP hard problems? **(10)**

* * * * *