Reg. No.										

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Institution of Manipal University

IV SEMESTER B.TECH. (COMPUTER SCIENCE & ENGINEERING)
END SEMESTER EXAMINATIONS, APRIL 2017
SUBJECT: DESIGN AND ANALYSIS OF ALGORITHMS [CSE 2202]
REVISED CREDIT SYSTEM
(21/04/2017)

Time: 3 Hours MAX. MARKS: 50

Instructions to Candidates:

- Answer **ALL** questions.
- Missing data may be suitably assumed.

1A. Explain the fundamental steps in algorithm design and analysis process with a neat diagram. **(3)**

1B. Design a recursive algorithm for computing 2^n for any nonnegative integer n that is based on the formula: $2^n = 2^{n-1} + 2^{n-1}$. Set up a recurrence relation for the number of additions made by this algorithm and solve it. Draw a tree of recursive calls for this algorithm. **(3)**

1C. Write selection sort algorithm to sort a list of n elements in ascending order by scanning the list to find its largest element and exchange it with the last element, putting the largest element in its final position. Then scan the list, to find the largest among the first $n-1$ elements and exchange it with the last but one element, putting the second largest element in its final position and then finding next largest element in first n-2 elements and putting in its final position and so on till the n^{th} largest element is in its proper position. Trace this algorithm for the list:

26, 54, 93, 17, 77, 31, 44, 55, 20 **(4)**

2A. For the graph shown in Fig. Q.2A, starting at vertex '*u*' and resolving ties by the vertex alphabetical order, traverse the graph by depth-first search and construct the corresponding depth-first search tree, showing all types of edges. Give the order in which the vertices were reached for the first time (pushed onto the traversal stack) and the order in which the vertices became dead ends (popped off the stack).

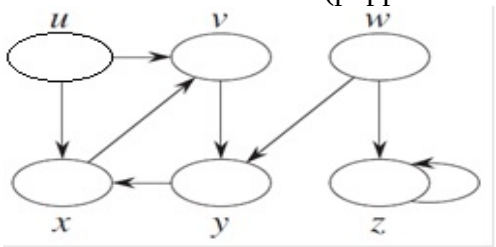

Fig. Q.2A **(4)**

2B. Write the algorithm for mergesort and trace the same on the following list to arrange in non-decreasing(ascending) order: ***5, 2, 4, 7, 1, 3, 2, 6.*** **(4)**

2C. Write nonrecursive binary search algorithm and analyse its worst case time efficiency. **(2)**

3A. What is 2-3 tree? Construct 2-3 tree for the list ***U, N, C, O, P, Y, R, I, G, H, T*** by successive insertion method starting from empty tree, considering alphabetical order. Show all stages. **(4)**

3B. Sort the list: ***5, 4, 9, 7, 19, 8, 17, 2, 6*** in non-increasing (descending) order using heapsort by clearly showing bottom-up heap construction and sorting stages. **(4)**

3C. Write comparison counting sort algorithm. What is the time efficiency of this algorithm? **(2)**

4A. Write the general procedure of Boyer-Moore string matching algorithm. Trace the same to search for a pattern: ***EARING*** in the text:

ENGINEERING_IS_SEARCHING_HEARING

Also find number of character comparisons made. **(4)**

4B. Write Floyd's algorithm and apply the Floyds algorithm to the graph shown in Fig.Q.4B showing all stages

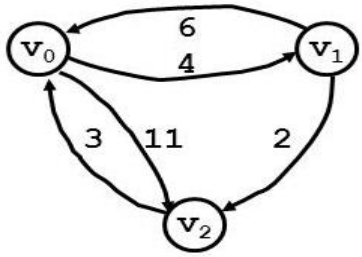

Fig.Q.4B **(3)**

4C. Apply the bottom-up dynamic programming algorithm to the following instance of the knapsack problem shown in Table Q.4C with capacity W=7 and find the optimal subset (Neatly show all the steps).

Table Q.4C

Item	Weight	Value
1	1	1
2	3	4
3	4	5
4	5	7

(3)

5A. Write the Huffman's algorithm and construct Huffman code for the following data given in Table Q.5A by clearly showing the construction of Huffman tree.

Table Q.5A

Character	A	B	C	D	E
Probability	0.1	0.1	0.3	0.25	0.25

(4)

5B. Apply the best-first branch-and-bound algorithm to the instance of the assignment problem given in the Table Q.5B, and find the optimal assignment of a person to a job. The table entries represents the assignment costs *C*[i, j] of assigning person '*i'* to job '*j'*. Clearly show the state space tree.

Table Q.5B

	Job *1*	Job *2*	Job *3*	Job *4*
Person *a*	8	1	4	6
Person *b*	3	5	9	4
Person *c*	11	8	2	3
Person *d*	2	4	7	7

(5)

5C. State and explain P and NP problems **(1)**