Reg. No.										

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

IV SEMESTER B.TECH. (COMPUTER SCIENCE & ENGINEERING)
END SEMESTER (MAKE UP) EXAMINATIONS, JUNE 2018
SUBJECT: DESIGN AND ANALYSIS OF ALGORITHMS [CSE 2202]
REVISED CREDIT SYSTEM
(14/06/2018)

Time: 3 Hours MAX. MARKS: 50

Instructions to Candidates:

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

1A. Define and explain different asymptotic notations along with example for each. **(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. Show the complete trace of the brute force string matching algorithm for the text string **NOBODY_NOTICED_HIM** and pattern string **NOT**. What is the basic operation of this algorithm? What happens to the analysis of this algorithm if the pattern string is present at the end of the text string or if the pattern string is not present in the text string? Give the proof for the same. **(4)**

2A. For the graph shown in Fig. Q.2A, starting at vertex 'a' and resolving ties by the vertex alphabetical order, traverse the graph by depth-first search and construct the corresponding depth-first search tree. 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).

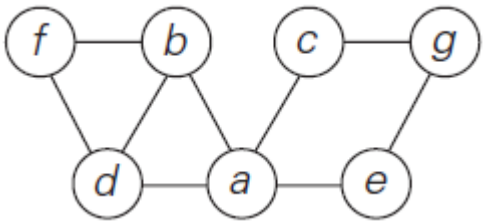

Fig. Q.2A **(4)**

2B. Write the algorithm for merge sort and trace the same on the following list to arrange in non-decreasing(ascending) order: **8, 3, 2, 9, 7, 1, 5, 4.** **(4)**

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

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

3B. Sort the list: S, O, R, T, I, N, G (in alphabetical 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.
Apply Horspool's algorithm to search for the pattern BAOBAB in the text
BESS KNEW ABOUT BAOBABS
Also find number of character comparisons made. **(4)**

4B. Write Kruskal's algorithm and apply Kruskal's algorithm to find a minimum spanning tree of the following graphs shown in Fig.Q.4B showing all stages

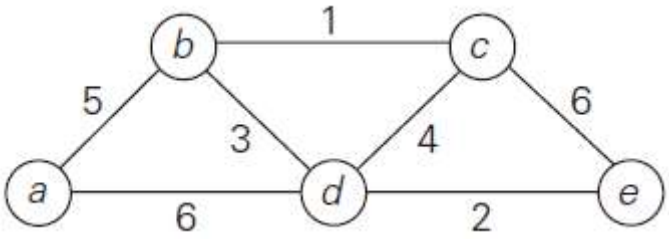

Fig.Q.4B **(3)**

4C. Apply the bottom-up dynamic programming algorithm to the following instance of the knapsack problem shown in Table Q.4C with capacity W=6 and find the optimal subset (Neatly show all the steps).

item	weight	value
1	3	$25
2	2	$20
3	1	$15
4	4	$40
5	5	$50

Table Q.4C

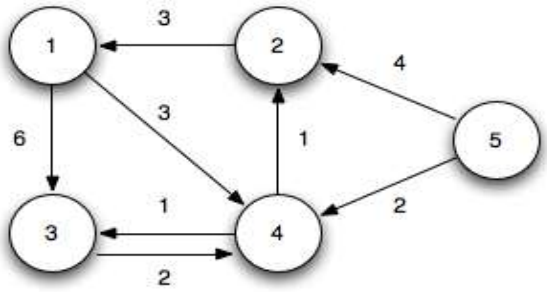

Fig Q.5A **(3)**

5A. Apply Warshall's algorithm to find the transitive closure for the above digraph shown in Q.5A by clearly showing all intermediate matrices. **(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.

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

Table Q.5B **(3)**

5C. The ***n-queens problem*** is to place *n* queens on an *n* × *n* chessboard so that no two queens attack each other by being in the same row or in the same column or on the same diagonal. Solve this for n = 4, by drawing the state-space-tree at each step. **(3)**