Reg. No. □□□□□□□□□□

Manipal Institute of Technology, Manipal

(A Constituent Institute of Manipal University)

V SEMESTER B.TECH (COMPUTER SCIENCE AND ENGINEERING)

END SEMESTER EXAMINATIONS, NOV/DEC 2015

SUBJECT: DESIGN AND ANALYSIS OF ALGORITHMS [CSE 303]

REVISED CREDIT SYSTEM
DATE: 27-11-2015

Time: 3 Hours MAX. MARKS: 50

Instructions to Candidates:

- Answer **ANY FIVE FULL** questions.
- Missing data, if any, may be suitably assumed.

1A. Write a pseudocode for consecutive integer checking algorithm for computing gcd(m, n) and also compute its time complexity. 2M

1B. Define Big-Theta notation. If $p(n) = a_k n^k + \ldots + a_1 n + a_0$ and $a_k > 0$, then prove that $p(n)$ Є $\Theta(n^k)$. 3M

1C. Design a recursive algorithm for computing 2^n for any positive integer n. Set up a recurrence relation and solve it. Draw a tree of recursive calls for this algorithm. 5M

2A. If the bubble sort algorithm makes no exchanges on its pass through a list, the list is sorted and the algorithm can be stopped. Design modified bubble sort algorithm that incorporates this improvement. 3M

2B. Trace the quicksort algorithm to sort the list **M, E, R, G, E, S, O, R, T** in alphabetical order (take first element as pivot). Also draw the tree of recursive calls made. Is quicksort stable sorting algorithm? 4M

2C. You are given an array of **n** distinct numbers with an unusual property: the numbers are strictly increasing from the first element to the **kth** element, for some unknown integer **k**, and the numbers are strictly decreasing from the **kth** element to the last element. Devise an **O(log n)** algorithm that receives such an array as an input and finds the maximum element in the array. For example, if the input array is: **1 4 7 8 6 3 0** then the output should be **8**. 3M

3A. Starting with vertex **A** and resolving ties by vertex alphabetical order perform the DFS and BFS traversal for the graph given in Fig. Q3A. Construct the corresponding DFS and BFS trees showing all kinds of edges. 4M

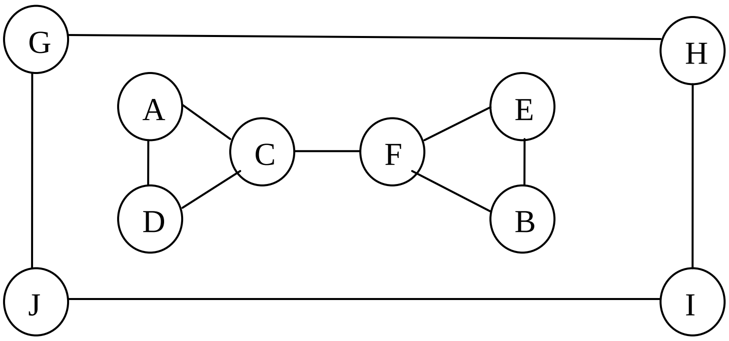


Fig. Q3A

3B. Design a presorting-based algorithm for computing a mode of an array $\mathbf{A[i], 0 \leq i \leq n-1}$ and determine its time complexity. 3M

3C. Sort the following list **S, O, R, T, I, N, G** (use the alphabetical order of the letters) by heapsort using the array representation of heaps. 3M

4A. Sort the following list **B, C, D, C, B, A, A, B** (whose values are known to come from the set **{A, B, C, D})** in alphabetical order by the distribution counting algorithm. 4M

4B. Consider a hash table with 9 slots. The hash function is h(k) = k mod 9. The collisions are resolved by chaining. The following 9 keys are inserted in the order: 5, 28, 19, 15, 20, 33, 12, 17, 10. Determine maximum, minimum, and average chain lengths in the hash table. 3M

4C. Let S and T be strings of lengths m and n respectively. Your task is to determine whether T is a sub-sequence of S, that is, whether the symbols of T occur in S in the same order as they appear in T, but not necessarily contiguously. For example, the string **grim** is a sub-sequence of the string **algorithm**, whereas the string **gram** is not. Design an **O(m + n)**-time algorithm to solve this problem. 3M

5A. Design a dynamic programming algorithm to compute binomial coefficient and determine its time complexity. 3M

5B. Determine the number of distinct minimum spanning trees using Kruskal's algorithm for the weighted graph given in Fig. Q5B. 3M

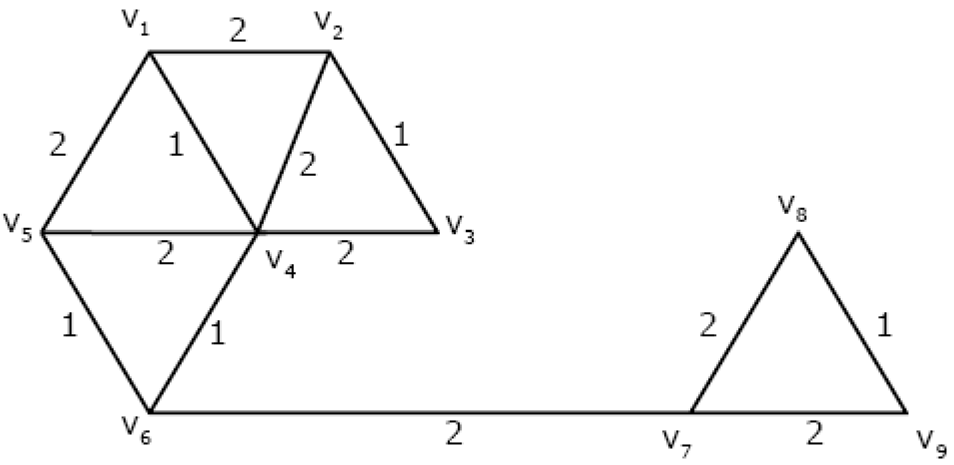
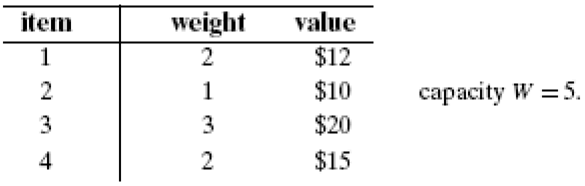


Fig. Q5B

5C. Apply the memory function algorithm for the instance of the knapsack problem given in the Table Q5C. 4M

Table Q5C



item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity $W = 5$.

6A. Write a pseudocode for the Huffman-tree construction algorithm. 3M

6B. Consider the directed graph given Fig. Q6B. There are multiple shortest paths between vertices S and T. Determine which one will be reported by Dijkstra's shortest path algorithm. 3M

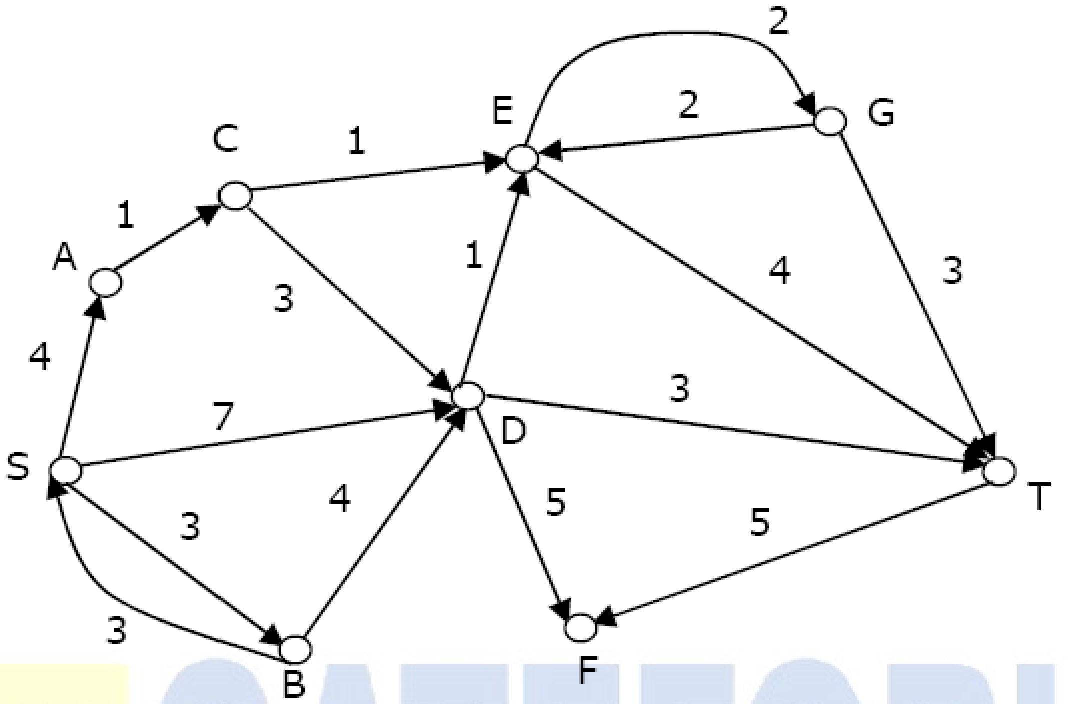


Fig. Q6B

6C. Define an nondeterministic algorithm. Design a nondeterministic algorithm to sort an array A[i], $1 \leq i \leq n$. Determine its time complexity and compare it with time complexity of deterministic sorting algorithm. 4M

****