Exam No: ________

GANPAT UNIVERSITY

M. Tech Sem. I (Computer Engineering) Regular Examination NOV/DEC – 2014

3CE103: COMPUTER ALGORITHMS

Max. Time: 3 hours **Max. Marks: 60**

Instructions:
1. This Question paper has two sections. Attempt each section in separate answer book.
2. Figure to the right indicates full marks.
3. Be precise and to the point in answering the descriptive questions.

SECTION – I

Q – 1 (a) Express complexity of following functions using theta (Θ) notation. Clearly indicates value of constants C1, C2 and n_0. **[6]**

1. $f(n) = 3n^2 - 6n$ 2. $f(n) = 2*6^n + 2n + 3$

(b) State whether the following statements are true or false. Justify the answer: **[4]**

1. $5 + n = O(n^2)$ 3. $n* \lg n + 8n^2 = \Omega(n^3)$
2. $\lg n + 2^n = O(n^3)$ 4. $n^2 + 6n + 7 = \Omega(n)$

OR

Q – 1 (a) Prove followings: **[6]**

1. $\log(\sqrt{n}) = O(\log n)$
2. If $P(n) = a_0 + a_1 n + a_2 n^2 + \ldots\ldots + a_m n^m$ then Prove that $P(n) = \Omega(n^m)$

(b) Express time complexity of following functions using omega (Ω) notation. Clearly indicates value of constants Cand n_0. **[4]**

1. $f(n) = 2^n + 5n^n$2. $F(n) = 1000$

Q – 2 (a) Solve following recurrences using master theorem. **[6]**

1. $T(n) = 16T(n/4) + n^2$ 2. $T(n) = 4T(n/2) + n^3$

(b) Solve the following recurrence and express the time complexity using theta notation : **[4]**

$T_n = 1$, if n=0

$= 3T_{n-1} + n + 2^n$, n>0

OR

Q – 2 (a) Solve the following recurrence and express the time complexity using theta notation : **[5]**

$T(n) = T(4n/5) + T(n/5) + n$

(b) Solve the following recurrence and express the time complexity using big-oh notation : **[5]**

$T(n) = T(n-1) + T(n-2)$, if n>1

$= n$, if n=0 or n=1

Q – 3 (a) Match the correct pairs for the following methods and their time complexities: **[2]**

1	Binary search (Best case)	(a)	O(n)
2	Bubble sort (worst case)	(b)	O(n log n)
3	Counting inversion (D & C)	(c)	O(1)
4	Insertion sort (Best case)	(d)	$O(n^2)$

(b) Write a function / algorithm to add two matrices of the size NxN. Find out the step count for this function using tabular method. **[4]**

(c) Give complexity of divide and combine stage of merge sort and show the working of merge sort on 10, 25, 18, -1, 87, 96, 34, 65. **[4]**

SECTION – II

Q – 4 (a) Find the longest common subsequence for the following two sequences using dynamic programming: **[5]**
X = PQRESQEP
Y = QSEPRQ

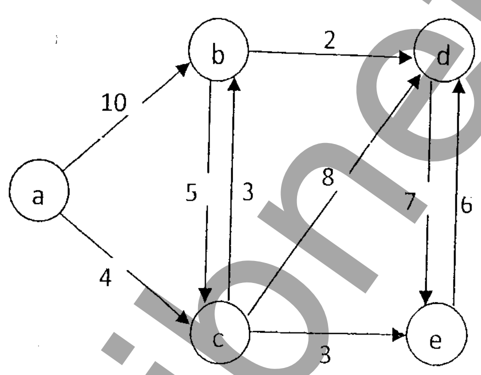
(b) Solve the following matrix chain multiplication problem using dynamic programming: **[5]**
A1: 20 x 5 A2: 5 x 50 A3: 50 x 10 A4: 10 x 30

OR

Q – 4 (a) On what kind of input the worst case of quick sort occurs? How it can be solved using randomized version of quick sort? **[5]**

(b) Write an algorithm to solve coin change problem using greedy design technique. Give its time complexity. And show the case of coin change problem in which greedy algorithm fails. **[5]**

Q – 5 (a) What is a minimum spanning tree? Find minimum spanning tree from following graph using prims' algorithm. Also write its time complexity. **[5]**



(b) Consider instance of the 0/1 knapsack problem as below with P depicting the value and W depicting the weight of each item whereas M denotes the total weight carrying capacity of the knapsack. Find optimal answer using greedy design technique. **[5]**
P = [40 10 5030 60] W = [8010 40 20 90] M = 110

OR

Q – 5 (a) Write bellman-ford algorithm for single source shortest path problem. Discuss its time complexity. **[5]**

(b) How backtracking strategy can beapplied to solve the subset sum problem? Let S = {1, 3, 4, 5, 6} and M=7. Find all possible subset of S, whose sum is equivalent to M. **[5]**

Q – 6 (a) Prove that Vertex-Cover $\equiv_P$ Independent-Set. **[5]**

(b) What do you mean by inversions in list? What is the complexity of counting inversion using brute force technique? Count the number of inversions on following data using divide and conquer method. And also indicates the time complexity of algorithm. **[5]**
99, 70, 25, 2, 5, 9, 14, 6, 3

----- END OF PAPER -----