Student Exam No._____

GANPAT UNIVERSITY

B. Tech. Semester VI (Computer Engineering / Information Technology)

Regular Examination May / June – 2013

2CE604 / 2IT604 : Design and Analysis of Algorithms

Time: 3 Hours **Total Marks: 70**

Instruction: 1. Figures to the right indicate full marks.
2. Each section should be written in a separate answer book.
3. Be precise and to the point in your answer.

Section - I

Q – 1 (a) Explain the following terms with example: [4]
1. Incorrect bound 2. Loose bound

(b) Show that max(f(n), g(n)) = Θ (f(n) + g(n)) [4]

(c) Prove the following: [4]
1. $f(n) = 5n^3 + 8n^2 + 3n + 2 = O(n^3)$
2. $f(n) = 9n^2 + 7n + 13 \neq \Omega(n^3)$

OR

Q – 1 (a) Show that $(n + a)^b = \Theta(n^b)$ [4]

(b) Prove the followings: [6]
1. $f(n) = 15\,n^2 - 5n = \Theta(n^2)$
2. $f(n) = 7n^2 + 2n + 5 \neq O(n)$
3. $f(n) = 2n^3 + 6n^2 + 7n = \Omega(n^2)$

(c) Discuss omega notation with graph. [2]

Q – 2 (a) Solve the recurrence: $T(n) - 3T(n-1) = n\,2^n$ [6]

(b) Solve the following homogeneous recursion: [3]

$$T(n) = \begin{cases} n & \text{, if } n = 0 \text{ or } 1 \\ T(n-1) + T(n-2) & \text{, otherwise} \end{cases}$$

(c) Solve the recurrence using master method: $T(n) = T(2n/3) + 1$ [2]

OR

Q – 2 (a) Solve using change of variable method: [6]
$T(n) = 4\,T(n/2) + n^2$

(b) Solve the following recurrence: [3]

$$T(n) = \begin{cases} 0 & \text{, if } n = 0 \\ 12 & \text{, if } n = 1 \\ 2T(n-1) + 3T(n-2) & \text{, otherwise} \end{cases}$$

(C) Write the general solution of the recurrence, when it has *k* distinct roots, each root with *m* multiplicity. [2]

Q – 3 (a) Solve the recurrence using Tree method: $T(n) = 4\,T(n/2) + n$ [3]

(b) Define space complexity. Discuss how recursion affects the space complexity. [3]

(c) Find the time complexity for the following function using profiling method: [4]

```
void ABC ( int n)
{   for ( i=0; i < n ; i++)
        for( j = i ; j >= 0 ; j++)
            printf ( "%d" , i * j);
}
```

(d) Explain empirical and theoretical approach to find time complexity. [2]

Section - II

Q – 4 (a) Explain divide and conquer method with merge sort algorithm. Give an example. **[6]**

(b) Explain activity scheduling problem with algorithm and solve following problem. Given a list of activities along with the starting and finishing time: **[6]**

Activity	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Start	1	2	3	4	7	8	9	9	11	12
Finish	3	5	4	7	10	9	11	13	12	14

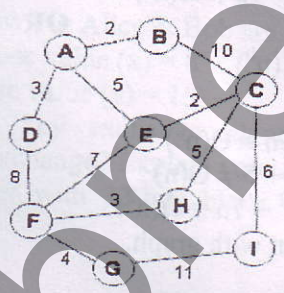
OR

Q – 4 (a) Write algorithm for comparison counting and show how comparison counting method sorts the list: 45, 2, 19, 10, 33, 22, 1, 23 **[6]**

(b) Compare normal exponentiation with fast exponentiation. Write algorithm for fast exponentiation. **[6]**

Q – 5 (a) Explain Dynamic programming concept. Using DP, solve following knapsack instance: **[6]**
Capacity = 4, [w1, w2, w3] = [1, 3, 2] and [p1, p2, p3] = [18, 16, 6]

(b) Write Prim's algorithm and Find Minimum Spanning tree for following graph using prim's method. **[5]**



OR

Q – 5 (a) What is greedy algorithm? Write down drawback of greedy algorithm and find out making change for 2051 using greedy strategy. The coins available are {500, 200, 50, 10, 2, 1} **[6]**

(b) How divide & conquer differ with Dynamic Programming? Solve binomial coefficient problem using dynamic programming. **[5]**

Q – 6 (a) What is branch and bound? How it differs from backtracking? Solve the assignment problem for following cost matrix to assign the jobs of 4 persons and draw the state space tree. **[6]**

	Job1	Job2	Job3	Job4
Person1	7	3	2	6
Person2	5	2	6	8
Person3	2	5	1	4
Person4	4	6	2	3

(b) Solve the matrix chain multiplication using dynamic programming: **[6]**
A1:13 x 4 A2:4 x 20 A3:20 x 3 A4:3 x 16 A5: 16 x 5

===== END OF PAPER =====