Exam No._____

GANPAT UNIVERSITY

B. TECH. SEM: V COMPUTER ENGINEERING/INFORMATION TECHNOLOGY
REGULAR EXAMINATION NOV-DEC 2016
2CE502/2IT502: DESIGN AND ANALYSIS OF ALGORITHMS

TIME: 3 HRS **TOTAL MARKS: 60**

Instructions: 1. Attempt all questions.
2. Figures to the right indicate full marks.
3. Each section should be written in a separate answer book.
4. Assume necessary data when needed.

SECTION - I

Q.1 **(A)** Define time complexity and calculate time complexity of given algorithm. **(4)**

```
Algorithm complexity(){
for(i=1;i<=n;i++){
  for(j=1;j<=n;j++){
    for(k=1;j<=200;k++){
      s:=0
  } } }
```

(B) Express time complexity of following functions using theta (Θ) notation. Clearly indicates value of constants C_1, C_2 and n_0. **(6)**

1) $f(n) = 8n + \lg n$ 2) $f(n) = 5\,2^n + 4n + 7$ 3) $f(n) = 2^n + 8n!$

OR

Q.1 **(A)** Analyze the following algorithm for best case time complexity using tabular method. **(4)**

```
void insertionSort(int arr[], int n){
  for (j = 2; j <= n; j++)  {
    key = arr[j];
    i = j-1;
    while (i > 0 && arr[i] > key){
      arr[i+1] = arr[i];
      i = i-1;
    }
    arr[i+1] = key;
} }
```

(B) Prove that $2^{n+1} = O(2^n)$ but $2^{2n} \neq O(2^n)$ **(2)**

(C) State whether followings are true or false. Justify your answer. **(4)**

1) $3n^3 + 4n \neq O(n^2)$ 2) $5n^2 + 3n + 7 \neq \Omega(n^3)$

Q.2 **(A)** Solve following recurrences and represent the time complexity using theta (Θ) notation. **(6)**

1) $T(n) = 4\,T(n/2) + n$ 2) $T(n) = T(n/3) + T(2n/3) + \Theta(n)$

(B) Solve the recurrence: $T_n = 3T_{n-1} + 4T_{n-2}$ with initial conditions $T_0 = 0$ and $T_1 = 5$ **(4)**

OR

Q.2 **(A)** Solve following recurrences and represent the time complexity using theta (Θ) notation. **(6)**

1) $T(n) = 2\,T(n/2) + n^2$ 2) $Tn = 2\,T_{n-1} + 1$ where $n >= 1$ and $T_0 = 0$

(B) Solve the recurrence: $T_n - 2\,T_{n-1} = (n+5)\,3^n$ where $n >= 1$ **(4)**

Q.3 (A) Define space complexity. Discuss how to calculate space complexity of recursive algorithm with example. (4)

(B) Differentiate following terms in detail. (4)
1) Big Omega and little Omega notation 2) Big Oh and little Oh notation

(C) Arrange the following growth rates in the increasing order. (2)
$O(n^3)$, $O(n \log n)$, $O(n (\log n)^2)$, $O(n^2 \log n)$, $O(40^n)$, $O(1)$

SECTION - II

Q.4 (A) Explain divide & conquer method using binary search algorithm. Write best and worst case time complexity of binary and sequential search algorithm. (4)

(B) Apply counting sort algorithm on given data and show the final content of count C array: 3, 3 ,5, 2, 4, 2, 0, 1, 0 (2)

(C) What is branch & bound technique? Explain it with example of assignment problem. (4)

OR

Q.4 (A) Discuss the best, average and worst case of quick sort. Illustrate the working of quick sort on given data: 17, 20, 21, 10, 8, 15, 89, 7 (4)

(B) Calculate total number of counting inversions for given input: 2,14,25,3,4,1,7,18 (2)

(C) Differentiate between class P and class NP problems. What is NP hard problem? (4)

Q.5 (A) What is minimum spanning tree? Apply prim's algorithm on graph shown in Figure 1 to construct minimum spanning tree. Consider node 0 as source node. (4)

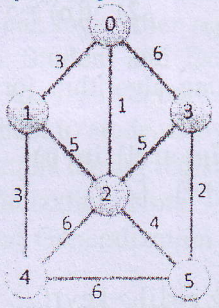

Figure 1

(B) For the following matrices find the order of parenthesization for the optimal chained multiplication using dynamic programming. (6)
Matrices: P: 5 x 4, Q: 4 x 6, R: 6 x 2, S: 2 x 7

OR

Q.5 (A) Write greedy algorithm for fractional knapsack problem. Apply it to solve fractional knapsack problem as shown in Table 1 where p is profit and w is weight of each item and M is knapsack capacity. Take M=30. (5)

Table 1

n	n_1	n_2	n_3	n_4
p_i	50	140	60	60
w_i	5	20	10	12

(B) What is memoization technique? Write fibonacci series algorithm using it. (5)

Q.6 (A) Explain breadth first search with example. (4)

(B) Discuss greedy algorithm for activity selection problem. (3)

(C) Apply dynamic programming algorithm to construct table for change of amount 12 with coins of denomination 1, 5, 6 and 9. (3)

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