Exam No: _____

GANPAT UNIVERSITY
M. TECH. SEM. – I COMPUTER ENGINEERING/INFORMATION TECHNOLOGY
REGULAR EXAMINATION JAN - 2013
3CE103/3IT103: COMPUTER ALGORITHMS

TIME:-3 HOURS] **[TOTAL MARKS: 70**

Instructions:

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)** Construct the Huffman code using greedy algorithm for following frequency table **[2]**

A:90, B:7, C:3, D:5, E:6, F:50, G:20

(B) Apply minimum finish time first, minimum starting time first and minimum interval first strategy on following Intervals and give your conclusion in term of which strategy returns optimal answer. Also mention time complexity for each strategy. **[4]**

I_1 (2-3), I_2 (0-5), I_3 (3-5), I_4 (5-7), I_5 (8-9), I_6 (4-6), I_7 (7-9), I_8 (1-10).

(C) Show overlapping of sub problem in recursive Fibonacci algorithm. Also write dynamic programming version of generating n^{th} Fibonacci number. **[6]**

OR

Q – 1 **(A)** What is memoization? Explain memoized version of Binomial Coefficient. **[6]**

(B) Prove that Vertex-Cover $\equiv_P$ Independent-Set **[6]**

Q – 2 **(A)** What is Clique in graph? Show that Clique of a graph is a NP problem. **[5]**

(B) Explain the case where greedy algorithm fails for making change problem. Also construct the dynamic programming table to give change of 10 with coins of denomination 1, 3, 5, and 7. Mention the conditions used to fill up the table. **[6]**

OR

Q – 2 **(A)** What is Hamiltonian cycle? Show that Hamiltonian cycle is a NP problem. **[5]**

(B) Write best case recurrence relation of Quick sort. Show the working of Quick sort on following input instance. Also mention nature of the input (i.e. Best Case, Worst Case, Average Case) **[6]**

10,15, 20, 84, 11, 59, 64, 3, 54

Q – 3 **(A)** Apply **Prim's** Algorithm on graph given in fig A (on page 3) and give minimum spanning tree. Also write the time complexity of **Prim**'s algorithm. **[6]**

(B) Draw the DFS and BFS tree for graph given in fig A (neglect the weight of edges). Also mention time complexity of both techniques. **[6]**

SECTION – II

Q – 4 **(A)** Let t_A (n) and t_B (n) denote the running times of two programs A and B respectively. For following pairs find the value of n for which program A is faster than program B. [4]

1.) $t_A(n) = 4n^3$, $t_B(n) = n^4$ **2.)** $t_A(n) = 100n$, $t_B(n) = 2n^3$

(B) Prove followings: [8]

1. $(n+a)^b = Ø(n^b)$ **2.** $(1/2)\,n^2 - 3n = Ø(n^2)$ **3.** $\text{Log}(\sqrt{n}) = O(\log n)$

OR

Q – 4 **(A)** Analyze following algorithm for its Best case and Worst case time complexity using tabular method. Represent the time complexity by Theta (Θ) notation. [6]

Algorithm SelectionSort (a , n)

```
//Here 'a' is the array having 'n' number of data
{
1.   for i:= 1 to n do{
2.          for j:= i+1 to n do{
3.                 if (a[i]>=a[j])
4.                        swap(a[i],a[j]); }}
}
```

(B) Explain the following terms with graph: [6]

1. Big-oh notation. 2. Omega notation. 3. Theta notation.

Q– 5 **Solve following recurrence relations and express your answer using big-oh (O) notation.**

(A) $T(n) = T(n/2) + n^2$ [3]

(B) $T(n) = 3T(n/4) + Θ(n^2)$ Use recurrence tree method only. [4]

(C) $T(n) = T(n-1) + n^2$ [4]

OR

Q – 5 **(A)** **Solve following recurrence relations and express your answer using big-oh (O) notation.** [5]

1.) $t_n = n$;if n<=1

$= 4t_{n-1} - 4t_{n-2} + 3^n$;Otherwise

2.) $T(n) = 4T(n/4) + n$

(B) **Write the recurrence relation (use tabular method) for following algorithm and solve it to represent complexity using Bog-oh (O) notation.** [6]

Algorithm Factorial (n) {

```
//this algorithm returns the factorial of number 'n'
1.   if (n=0 or n=1)
2.          return 1;
     else
3.          return (n*factorial(n-1)); }
```

Q – 6 **(A)** Consider various instance of the **Fractional** knapsack problem as below, with v depicting the value and w depicting the weight of each item whereas W depicting the total weight carrying capacity of the knapsack **[6]**

i = [1 2 3] v = [25 10 35] w = [10 15 25] W = 30

i = [1 2 3] v = [100 45 55] w = [70 25 50] W = 100

apply

1. Minimum weight first criterion
2. Maximum value first criterion
3. Maximum value density first criterion to the above inputs & give vital conclusion based on answer

(B) Solve the following Assignment Problem (i.e. find the optimal assignment value) using Branch and Bound technique. **[6]**

	1	2	3	4
P	10	2	7	11
Q	5	9	8	9
R	4	4	5	15
S	2	12	9	6

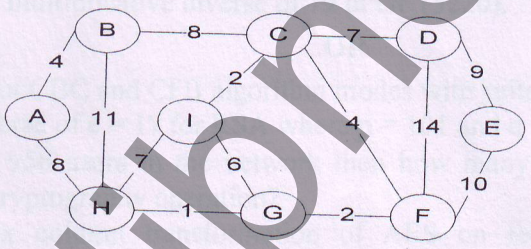

Fig A

END OF PAPER