(Following Paper ID and Roll No. to be filled in your Answer Book)

PAPER ID : 2165 Roll No. | | | | | | | | | | |

B. Tech.

(SEM. V) THEORY EXAMINATION 2011-12

DESIGN AND ANALYSIS OF ALGORITHMS

Time : 3 Hours *Total Marks : 100*

Note :– Attempt **all** questions.

1. Attempt any **four** parts of the following :

   (a) If $f(n) = a_m n^m + \ldots\ldots + a_1 n + a_0$ and $a_m > 0$, then show that $f(n) = \Omega(n^m)$.

   (b) Consider the following function :

```
int RSUM(int A [], int n)
{
    if (n) return RSUM (A, n-1) + A [n-1];
    return 0;
}
```

   Determine the asymptotic complexity of the function RSUM.

   (c) Consider a polynomial

   $P(x) = (\ldots\ldots(c_n * x + c_{n-1}) * x + c_{n-2}) * x + c_{n-3} * x \ldots\ldots) * x + c_0$.

   Estimate the time complexity to evaluate the polynomial $P(x)$.

(d) Show that following equalities are incorrect :

(i) $n^2 \log n = \Theta(n^2)$

(ii) $n^2 / \log n = \Theta(n^2)$.

(e) Sort the elements of the given array A using shell sort algorithm :

A = [20, 35, 18, 8, 14, 41,3, 39]

(f) Prove that Quick sort algorithm takes O (n^2) time to sort an array of n elements in the worst case.

2. Attempt any **two** parts of the following :

(a) Define a red-black tree. Let h be the height of a red-black tree and let n be the number of internal nodes in the tree. Then show that :

$$h \leq 2 \log_2 (n+1)$$

(b) Consider T is a B-tree of order m and height h. Let d = m/2 and let n be the number of elements in T. Then show that :

$$\log_m (n+1) \leq h \leq \log_d ((n+1)/2) + 1.$$

(c) What is a Fibonacci heap ? Discuss the applications of Fibonacci heaps.

3. Attempt any **two** parts of the following :

(a) Let T(n) = a T (n/b) +c, then show that T(n) = O (log n), if a = 1, otherwise T(n) = O (n $\log_b$a).

(b) Give the high-level description of Kruskal's algorithm to find the minimum cost spanning tree of an n-vertex undirected network.

(c) What is an optimization problem ? How are greedy method can be used to solve the optimization problem ?

4. Attempt any **two** parts of the following :

(a) Explain dynamic programming method. Formulate dynamic programming recurrence equation for 0/1 knapsack problem.

(b) Describe travelling salesman problem (TSP). Show that a TSP can be solved using backtracking method in the exponential time.

(c) What is the graph coloring problem ? What do you mean by optimal coloring of a graph ? Show that every bipartite graph is 2-colorable.

5. Write short notes on any **four** parts of the following :

(a) NP completeness

(b) Approximation algorithms

(c) Fast Fourier transform

(d) String matching algorithms

(e) N-queen problem

(f) Hamiltonian cycles.