Name : .....

Roll No. : .....

Invigilator's Signature : .....

CS/B.Tech(CSE)/SEM-5/CS-503/2010-11

2010-11

DESIGN & ANALYSIS OF ALGORITHMS

Time Allotted : 3 Hours *Full Marks* : 70

The figures in the margin indicate full marks.

Candidates are required to give their answers in their own words as far as practicable.

GROUP – A
(Multiple Choice Type Questions)

1. Choose the correct alternatives for the following : $10 \times 1 = 10$

   i) Which of the following problems is solved by using Branch and Bound method ?

      a) Knapsack problem

      b) Hamiltonian problem

      c) Travelling salesman problem

      d) 15 puzzle problem.

   ii) Which of the following algorithm design technique is used in the quick sort algorithm ?

      a) Dynamic programing b) Backtracking

      c) Divide and conquer d) Greedy method.

   iii) 'Small o' of g (n) is

      a) asymptotically loose b) asymptotically tight

      c) same as 'Big O' d) none of these.

iv) The time factor when determining the efficiency of an algorithm is measured by

a) counting microseconds

b) counting the number of key operations

c) counting the number of statements

d) counting the kilobyte of algorithm.

v) Which of the following is used to depict the working of algorithm ?

a) Flowchart b) Pseudo code

c) Source code d) All of these.

vi) The complexity of Bubble sort is

a) O (n) b) O (log n)

c) $O(n^2)$ d) O (n log n).

vii) Which one is true of the following ?

a) All NP hard problems are NP complete

b) All NP complete problems are NP hard

c) Some NP complete problems are NP hard

d) None of these.

viii) BFS on a graph G = (V, E) has running time

a) O (|V| + |E|) b) O (|V|)

c) O (|E|) d) none of these.

ix) The fractional Knapsack problem can be solved by using

a) Greedy method

b) Divide and Conquer method

c) Dynamic programming

d) none of these.

x) Time complexity of Binary Search algorithm on n items is

a) O (n) b) O (n log n)

c) $O(n^2)$ d) O (n log n).

GROUP – B
(Short Answer Type Questions)

Answer any *three* of the following. $3 \times 5 = 15$

2. Let $A[1..n]$ be a sorted array of n distinct integers. Give a divide-and-conquer algorithm that can find an index i such that $A[i] = i$ (if one exists) with running time $O(\log n)$.

3. What do you mean by dynamic programming ? Write the algorithm of matrix-chain multiplication.

4. Write the FFT algorithnm and find the computational complexity of this algorithm.

5. Write an algorithm of eight queens problem and find time complexity of the algorithm. Explain the algorithm using an example.

6. Show that 2SAT is in P but 3SAT is NP-complete.

GROUP – C
(Long Answer Type Questions)

Answer any *three* of the following. $3 \times 15 = 45$

7. Suppose we have a recurrence relation $T(n) = aT\left(\frac{n}{b}\right) + f(n)$. Show that the followings :

   a) If $af\left(\frac{n}{b}\right) = kf(n)$ for some constant $k < 1$, then $T(n) = O(f(n))$.

   b) If $af\left(\frac{n}{b}\right) = kf(n)$ for some constant $k > 1$, then $T(n) = O\left(n^{\log_b a}\right)$.

   c) If $af\left(\frac{n}{b}\right) = kf(n)$ for some constant $k = 1$, then $T(n) = O(f(n) \log_b n)$.

8. What is Heap property ? Write an algorithm to make a Heap containing elements. Then, show that how can you insert an element into a Heap. Then, write the algorithm of Heap sort and find the running time of this algorithm. Write an algorithm to find the existence of an element into a Heap.

9. Write the algorithm of Quick sort. Then find the best case, worst case and average case time complexities of this algorithm. You also find the kth smallest element of n elements with time complexity less than that of Quick sort algorithm.

10. Using any method you like, compute the following subgraphs for the weighted graph below :

    a) a depth-first search tree, starting at the top vertex ;

    b) a breadth-first search tree, starting at the top vertex ;

    c) the minimum spanning tree.

11. Answer any *three* of the following :

    a) Show that 3-colouring problem is *NP*-complete.

    b) Write the travelling salesman problem with an algorithm.

    c) What is 0/1 knapsack problem ? Explain it with an example.

    d) How can you convert the following recursion algorithm (Algorithm 1) to tail recursion ?

Algorithm 1 *Fib* (n)

```
i)   if n ≤ 1 then
ii)      Return ( n )
iii) else
iv)      Return Fib ( n − 1 ) + Fib ( n − 2 )
v)   end if
```