Register No: ………………………

M.C.A DEGREE EXAMINATIONS: JUNE 2011

Second Semester

MASTER OF COMPUTER APPLICATIONS

MCA507: Design and Analysis of Algorithms

Time: Three Hours **Maximum Marks: 100**

Answer ALL Questions:-

PART A (10 x 2 = 20 Marks)

1. Define an algorithm

2. Define abstract data type

3. Define time complexity and space complexity of an algorithm

4. Calculate the Big O notation for the following

   i) $7\log_2(n)+4n$ ii) $5n^5+n\log_2(n)$

5. What is the running time of divide and conquer algorithm?

6. Why did we not include multiplications by 10^n in the multiplication count M(n) of the large integer multiplication algorithm?

7. What is the principle difference between dynamic programming and divide and conquer technique?

8. Say True or False

   a) A sequence of values in a row of the dynamic programming table for an instance of the knapsack problem is always non-decreasing

   b) A sequence of values in a column of the dynamic programming table for an instance of the knapsack problem is always non-decreasing

9. Define subset problem

10. What data structure would you use to keep track of live nodes in a best-first branch-and-bound algorithm?

PART B (5 x 16 = 80 Marks)

11. a) What are the fundamental characteristics of an algorithmic problem solving?

(OR)

b) (i) Explain any three important problem types.

(ii) Let A be the adjacency matrix of an undirected graph. Explain what property of the matrix indicates that i) the graph is complete ii) the graph has a loop, i.e., an edge connecting a vertex to itself, iii) the graph has an isolated vertex, i.e., a vertex with no edges incident upon it.

12. a) (i) Explain the asymptotic notations that are used to analyze the algorithms.

(ii) Describe the relations between asymptotic notations with a neat sketch.

(OR)

b) (i) Find the order of growth of $i)\sum_{i=0}^{n-1}\left(i^2+1\right)^2$ and $ii)\sum_{i=2}^{n-1}\left(\log i^2\right)$

(ii) Consider the following algorithm

```
ALGORITHM Mystery(n)
//Input: A nonnegative integer n
 S←0
for i ← 1 to n do
      S←S + i * i
return S
```

(i) What does this algorithm compute?

(ii) What is its basic operation?

(iii) How many times is the basic operation executed?

(iv) What is the efficiency class of this algorithm?

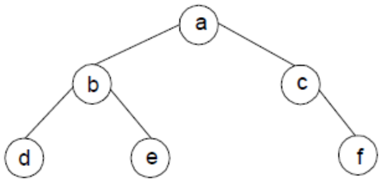
Suggest an improvement or a better algorithm and indicate its efficiency class.

13. a) Describe the divide and conquer methodology. How is it applied in binary search? Give the algorithm. Do the worst case analysis for binary search algorithm.

(OR)

b) (i) Design a divide and conquer algorithm for computing the number of levels in a binary tree. What is the efficiency class of the algorithm?

(ii) Explain binary tree traversals. Traverse the following binary tree and show the contents of the traversal's stack as the algorithm progresses:



14. a) (i) Explain the method of finding c(n, k) using dynamic programming approach.

(ii) Compute c (6, 3) by applying this method. Is it possible to compute c (n, k) by filling the dynamic programming table column by column rather than row by row?

(OR)

b) (i) What is an optimal binary search tree?

(ii) Write an algorithm to find optimal binary search tree. Also mention its space and Time efficiency.

15. a) Define Hamiltonian circuits. What is the difference between Hamiltonian circuit and Traveling salesman problem? Explain.

(OR)

b) (i) Explain Branch-and-Bound methodology.

(ii) Solve the following instance of the knapsack problem by the branch-and-bound algorithm.

Item	**Weight**	**Value**
1	10	100
2	7	63
3	8	56
4	4	12

Total weight (W) = 16

****