Minor Test I – September 2012

**MCA301: Design and Analysis of Algorithms**

# Time : 1 hr Max Marks : 20

1. Give a pair of functions f(n) and g(n) such that neither f(n) = O(g(n)) nor g(n) = O(f(n)).

 2

1. Can you use Master’s theorem to solve the following recurrence
	1. T(n) = 4T(n/2) + n^(1.5)?
	2. T(n) = 2T(n/2) + n /log^2 n?

Justify your answer.

 4

1. Solve the following recurrence relation iteratively

 T(n) = T(4n/5) + T(n/5) + n

 T(1) = 1

 OR

Show that 2n^2 + 3n – 5 = O(n^3 – 3n^2 – 4n – 9)

 4

1. Let x be an array of numbers. Consider the following function to compute the maximum of n numbers. Write a recurrence relation for its time complexity:

 maximum( x,n)

 If n == 1 return x[1];

 max = maximum(x, n - 1);

 If x[n] > max

 return x[n];

 else return max;

 3

1. Discuss and compare the worst case time complexity of the following two implementations of insertion sort:
	1. Sequential search is used to insert x\_i.
	2. Binary search is used to insert x\_i.

 2

1. Given a set S of n integers, write an O(n log n) time algorithm that determines, for a given integer x, whether there is a pair of elements in S whose sum is x. Give the step by step analysis in 2-3 lines.

 5