DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
FIRST SESSIONAL
SUBJECT: (IT 509) Design And Analysis of Algorithm

Examination	**: B.TECH Semester - V**	**Seat No.**	**:**
Date	**: 05/08/2016**	**Day**	**: Friday**
Time	**: 11:30 – 12:45**	**Max. Marks**	**: 36**

INSTRUCTIONS:
1. Figures to the right indicate maximum marks for that question.
2. The symbols used carry their usual meanings.
3. Assume any necessary data but giving proper justifications.
4. Be precise, clear and to the point in answering the questions. Unnecessary elaborations will not fetch more marks.

Q.1 Do as directed. **[12]**

(a) The minimum number of comparisons required to determine if an integer appears more than n/2 times in a sorted array of n integers is **[1]**

(A) $\Theta(n)$ (B) $\Theta(\log n)$ (C) $\Theta(\log^* n)$ (D) $\Theta(1)$

(b) Solve the recurrence relation and give your answer in Big-O, **[2]**
$T(n) = n * T(n-1)$; if $n > 1$ and $T(n) = 1$; if $n = 1$.

(c) Design the recurrence relation for the following code snippet and solve it: **[2]**

```
Sessional_1(a, n){
    if(1 == n)
        return a;
    m = n/2;
    return Sessional_1(a, m) * Sessional_1(a, n-m);
}
```

(d) Consider an input that is an array of **N** numbers. The problem is to check whether this array is distinct or not i.e. if ***no*** two numbers are the same, the answer is YES, otherwise the answer is NO. Show that this problem can be solved in Θ (N lgN) time. **[3]**

(e) Solve the following recurrence relations: **[3]**
(1) $T(n) = T(\sqrt{n}) + 1$ (2) $T(n) = 4\,T\left(\frac{n}{3}\right) + n^2$

(f) Compare the following two expression and tell which one is asymtotically smaller then other **or** both are asymptotically equal: (1) $n^{1+1/logn}$ (2) $n\,loglogn$ **[1]**

Q.2 Attempt ***Any TWO*** of the following questions. **[12]**

(a) Write the MIN-MAX algorithm using Divide and Conquer paradigm. Write down the recurrence equation and derive the formula for minimum comparison require for finding minimum and maximum from given array. **[6]**

(b) Write the Divide and Conquer Binary Exponent (power) algorithm for find a^n, derive its complexity and compare it with conventional method to do so. **[6]**

(c) Consider the variation of the binary search algorithm that splits the input into three sets of equal sizes (almost) let's call it ternary search. Write down the algorithm for ternary search and analyze its complexity. Is binary search is preferable over ternary search? Yes/No with proper justification. **[6]**

Q.3 (a) Solve the recurrence equation given below, **[6]**

$$t_n = \begin{cases} 0\,; if\ n = 0 \\ 2t_{n-1} + n + 2^n\,; Otherwise \end{cases}$$

(b) Analyze the time complexity of Quicksort algorithm's Best case, Average case, and Worst case in detail. It is possible to eliminate worst case always if we use random element as pivot in Quicksort? Yes/No with proper justification. **[6]**

OR

Q.3 (a) Assuming that the partitioning in Quicksort produces a uniform 9 to 1 (9:1) proportional split, analyze the time complexity of the quick sort with recursion-tree method. Is this partitioning closer to the worst-case or best-case? Reason your answer. **[6]**

(b) Write the Median of Median (MOM) algorithm, and explain how it can be useful for finding the median order statistics in linear (O(N)) time. Analyze and prove its complexity with logical argument. **[6]**