Register No.: | 1 | G | C | A | 4 | 3 | |

Department of Mathematical and Computational Sciences
National Institute of Technology Karnataka, Surathkal

even Semester (2017- 2018)
Course Code: MCA801
Date:08/02/2018

Time:10.30 AM to 12.00 AM

Examination: Mid Sem
Course Name: Computer Algorithms
Maximum Marks: 50

INSTRUCTIONS:
1. Answer ALL questions.
2. Rough work should NOT be done anywhere on the Question Paper.

Q.1. (a) Prove that $\sum_0^n 2^i = 2^{n+1} - 1$ by mathematical induction. [05]

(b) Consider the following list of functions and arrange them in ascending order of their growth rate using Big-oh notation. $g_1(n) = 2^{\sqrt{(\log n)}}$, $g_2(n) = 2^n$, $g_3(n) = n^{4/3}$, $g_4(n) = n(\log n)^3$, $g_5(n) = n^{\log n}$, $g_6(n) = 2^{2^n}$, $g_7(n) = 2^{n^2}$. [05]

Q.2. (a) Suppose you have four algorithms with the running time listed below. These running times are the exact number of instructions performed as a function of the input size n. Suppose you have a computer that can perform 10^{10} instructions per second. For each of the algorithms, what is the largest input size n for which you would be able to get the result within an hour? [05]

i. n^2
ii. n^3
iii. $100n^2$
iv. 2^n
v. $n \log n$

(b) Find the total number of operation performed by the following code. Express the same in Θ notation. [5]

```
Algorithm Exponentiate(x, n)
{
m = n; power = 1; z = x;
while(m > 0) do
{
    while(m  mod 2 == 0) do
    {
        m = ⌊(m/2)⌋; z = z * z;
    }
    m = m − 1; power = power * z;
} return power;
}
```

Q.3. (a) Solve the following recurrence using recursion tree method [04]

$$T(n) = \begin{cases} 4T(n/2) + n^3 & \text{if } n > 1 \\ 1 & \text{if } n \le 1 \end{cases}$$

(b) Find the asymptotic tight bound for [06]

i. $T(n) = 3T(n/2) + n \log n$

ii. $T(n) = 2T(n/4) + n$

iii. $T(n) = T(\sqrt{n}) + \log n$

Q.4. (a) Let $x[1..n]$ and $y[1..n]$ contain 2 sets of integers, each sorted in nondecreasing order. Write an algorithm that finds the median of the $2n$ combined elements in $O(\log n)$ time. [04]

(b) Write an algorithm PARTITION(A, p, r), which partitions a given array of n elements around the element $A[r]$. Parameters p and r are the start and end index respectively in array A. Trace your algorithm for the data $\langle 5, 10, 12, 16, 2, 1, 15, 8, 6\rangle$ [06]

Q.5. (a) Suppose we modify the deterministic version of the quick-sort algorithm so that, instead of selecting the last element in an n-element sequence as the pivot, we choose the element at index $\lfloor n/2 \rfloor$, that is an element in the middle of the sequence. Analyze the running time of this version of quick-sort on a sequence that is already sorted? [06]

(b) For the modified quick-sort algorithm, in the question above, describe the kind of sequence that would result in $\Theta(n^2)$ running time. [04]

(*This question paper contains* 2 *page(s) and* 5 *Questions.*)