SPRING MID SEMESTER EXAMINATION-2017

Design & Analysis of Algorithms

[CS-3001]

Full Marks: 25 **Time: 2 Hours**

Answer any five questions including question No.1 which is compulsory.
The figures in the margin indicate full marks.
Candidates are required to give their answers in their own words as far as practicable and all parts of a question should be answered at one place only.

Q1 Answer the following questions: (1 x 5)

a) Consider the following C function.

```
int FUN( int x, int n)
{
    if ( n==0)
        return 1;
    else if(n==1)
        return x;
    else if (n%2==0)
            return FUN( x * x, n/2);
    else
           return FUN( x * x, n/2 ) * x ;
}
```

i) Write down the recurrence T(n) for the above function **FUN(int x, int n)**.

ii) Derive the worst case Time Complexity of **FUN(int x, int n)**

b) Do as directed.

i) State true or false.

$f(n) = O(f(n))$

ii) Fill in the blanks.

_____ is the best case Time Complexity to find out the smallest element in a binary MAX-HEAP containing **n** numbers.

c) What is the solution of the following recurrence?

$T(n) = 9T(n/3) + n^2 \log n$

d) Consider a complete binary tree where the left and the right sub-trees of the root are MIN-HEAPs. What is upper bound Time Complexity to convert the tree into a MIN-HEAP?

e) Match the following Algorithms with its best case Time Complexity

Insertion Sort	$O(n^2)$
Quick-sort	O(log n)
Simple Bubble Sort	O(n)
Binary-Search	O(n log n)

Q2 a) Describe the asymptotic efficiency of Algorithms with suitable examples. (2.5)

b) Solve the following recurrences. (2.5)

a) $T(n) = 3T(n/2) + n$ where $n>1$ and $T(1)=1$

b) $T(n) = 2T(n-1) + 1/n$, $T(0) = 1$

Q3 a) Given an unsorted array A[1..n], where odd indexed elements are sorted in ascending order and the even indexed elements are sorted in descending order. Design an Algorithm to sort the array in O(n) worst-case time, in ascending order. (2.5)

b) Write the algorithm to build a MAX-HEAP? Describe your Algorithm to build a MAX-HEAP from the array A = {5, 7, 8, 2, 1, 0, 3, 9, 4, 5, 6} in a step by step process. Derive the time complexity of building a MAX-HEAP. (2.5)

Q4 a) Write the recursive Binary-Search Algorithm. Find out its recurrence and discuss its best case and worst case time complexities. Also identify the lines of the Algorithms that support each part of Divide, Conquer and Combine paradigm respectively. (2.5)

b) Write the PARTITION() Algorithm of Quick-Sort. Describe in a step by step process to get the pass1 result of PARTITION() by taking last element as pivot on the following array elements. (2.5)

13, 12, 10, 18, 14, 20, 11, 18, 14, 15, 15

Describe the Time complexity of PARTITION() Algorithm.

Q5 a) Write an Algorithm MAX-MIN(A, max, min) to find out the maximum and minimum elements of an array A by using Divide-and-Conquer strategy. Compare this with the STRAIGHT-MAX-MIN(A, max, min) Algorithm in terms of number of comparisions. (2.5)

b) Write the Algorithm for the procedure HEAP-EXTRACT-MAX(A), where the procedure removes and returns the element of MAX-HEAP with largest key. Illustrate the operation of HEAP-EXTRACT-MAX on the Heap A={1, 3, 2, 6, 4, 5, 7, 8, 9} (2.5)

Q6 a) Given a set S of **n** integers and another integer **x**, determine whether or not there exist two elements in S whose sum is exactly **x**. Describe a Θ(nlogn) time Algorithm for the above problem. (2.5)

b) Explain the Master Theorem with suitable examples. (2.5)

==========XXXXXXXX==========

ALL THE BEST