SPRING RE-MID SEMESTER EXAMINATION-2018

Design & Analysis of Algorithms

[CS-2008]

Full Marks: 25 **Time: 1.5 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 function

```
int FUN(int n)
{
   int i, j, k = 0;
   for (i  = n/2; i <= n; i++)
      for (j = 2; j <= n; j = j * 2)
            k = k + n/2;
   return k;
}
```

What is the returned value of the above function?

b) 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?

c) What is the difference between Divide & Conquer and Greedy algorithm?

d) Write the merge sort procedure (only) which divides the array into two parts such that first part contains elements twice of second part. Also derive the time complexity of that merge sort.

e) We want to store files of lengths (in MB) {12, 34, 55, 73,24, 11, 34, 56, 78, 91, 34, 38, 45} on three tapes T1, T2 & T3. How should we store them on the three tapes so that the mean retrieval time is minimized?

Q2 (5)

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

Q3 (5)

Find an optimal Huffman code for the following set of frequencies
a:2 b:1 c:6 d:3 e:5 f:2

Q4 (5)

Insertion sort can be expressed as a recursive procedure as follows. In order to sort A[1…n], we recursively sort A[1..n-1] and then insert A[n] into the sorted array A[1..n-1]. Write the procedure and a recurrence for the running time of this recursive version of insertion sort.

Q5 (5)

Given 12 activities, A=< a_1, a_2,….,a_{10},a_{11},a_{12} > along with their start time (s_i) and finish time (f_i) are given as follows:

i	1	2	3	4	5	6	7	8	9	10	11	12
s_i	44	7	37	83	27	49	16	44	44	58	27	26
f_i	86	25	96	89	84	62	17	70	84	94	79	57

Use an efficient method that computes a schedule with largest number of activities on that stage.

(5)

Q6

Solve the following recurrences

a) $T(n) = 4T(n/3)+n^2\log n$

b) $T(n) =T(\sqrt{n}) + \log_2 n$

Q7 (5)

Illustrate the level order traversal of the Max-Priority queue after inserting (Using HEAP-INSERT() procedure) two elements 151 and 46 in the order into the given queue A = < 95, 89, 82, 50, 80, 21, 78, 17, 25 >.

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