SPRING MID SEMESTER EXAMINATION-2019

Design & Analysis of Algorithms

[CS-2008]

Full Marks: 20 **Time: 1.5 Hours**

Answer any four 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) Define Ω–Notation.

b)
```
int fun ()
{
    int i, s = 0;
    for (i  = 1; i ≤1024; i=2*i)
        s=s+i;
    return s;
 }
```
What is the time complexity of the above function fun?
A) Θ(1) B) Θ(log n) C) Θ(n) D) Θ(nlog n)

c)
```
int fun1(int n)
{
    int i, j, k = 0;
    for (i  = n; i ≥ 1; i=i/2)
        for (j = 1; j ≤ i; j++)
            k = k + 1/n;
    return k;
 }
```
What is the returned value of the above function fun1?
A) Θ(1) B) Θ(log n) C) Θ(n) D) Θ(nlog n)

d) The performance of merge sort depends on which type of data?
   A) Increasing order data
   B) Decreasing order data
   C) 50% data are increasing and 50% data are decreasing
   D) Does not depend upon the data set.

e) Find out the min-heap (show in diagram) evolved, after inserting 2 and 0 in that order into the heap={1, 3, 2, 4, 5, 6, 8, 9, 8, 7}.

Q2 Write a sorting algorithm for the following elements stored in a n-element array A such that the execution time will be least. The array A is given as, A ={1, 2, 3, 4, 5, 6, …….., 99, 100 }, if n=100. Find the execution time in asymptotic notation. (5)

Q3 Solve the following recurrences (5)

a) $T(n) = 2T(\sqrt{n}) + 1$, $T(1)=1$

b) $T(n) = 4T(n/2) + n \log_2 n$, $T(1)=1$

Q4 In a social gathering, there are m boys and n girls (m > n) of different ages. You have two unsorted arrays giving their ages (one for the boys, the other for the girls). Devise an efficient O(m log n) algorithm to find out the ages that are common between both the boys and girls. (5)

Q5 Write an algorithm MAX-HEAP-CHANGE-KEY(A, n, i, key), to re-build a n-element max-heap A, after the value at node with index i has been changed to a new value key. Illustrate the operation of MAX-HEAP-CHANGE-KEY(A, 12, 2, 2) on the heap A={15, 13, 9, 5, 12, 8, 7, 4, 0, 6, 2, 1}. Assume root is at index 1. (5)

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