Semester: V (Regular)
Sub & Code: DAA, CS-3001
Branch (s): CSE & IT

AUTUMN MID SEMESTER EXAMINATION-2016

Design & Analysis of Algorithm

[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 n)
{
  if (n<=2)
    return 1;
  else
    return  fun(√n);
}
```

Represent this function by recurrence and find out the time complexity.

b) Consider the following pairs of functions f(n), g(n). Which pair the functions are such, that f(n) is O(g(n)) and g(n) is not O(f(n)) ?

A. $f(n)=n^3$, $g(n) = n^2\log(n^2)$ B. $f(n)=\log(n)$ $g(n) = 10 \log n$

C. $f(n)=1$, $g(n) = \log n$ D. $f(n)=n^2$, $g(n) = 10n \log n$

c) Which of the following sorting algorithms in its typical implementation gives best performance when applied on an array which is sorted, or almost sorted (at most two elements are misplaced).

A. Quick Sort B. Heap Sort C. Merge Sort D. Insertion Sort

d) A priority queue is implemented as a Max-Heap. Initially, it has 5 elements. The level order traversal of heap is as 11, 9, 6, 4, 3. Two new elements 2 and 8 and inserted in the heap in that order. Find out the level order traversal of the heap after the insertion of the elements?

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

Q2 a) What is purpose of algorithm analysis? Find out the complexity of the given function $\sum_{k=0}^{n-1} 2^k$. (2.5)

b) Solve the following recurrence. (2.5)
$T(n) = \sqrt{n}\, T(\sqrt{n}) + n$
Assume that T(n) is constant for sufficiently small n.

Q3 a) Solve the following recurrence. (2.5)
$T(n) = 3\, T(n/4) + n^2$
Assume that T(n) is constant for sufficiently small n.

b) Write a function in C language to arrange the numbers stored in an array as follows: odd numbers followed by even numbers, in O(n) time. The prototype of the function is given as void ARRANGE-ARRAY(int [], int); (2.5)

Q4 a) Write down the merge sort algorithm that uses the merge procedure. Describe the time complexity of each terms of the divide, conquer and combine strategy. (2.5)

b) 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: (2.5)

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.

Q5 a) Write the algorithm for MAX-HEAPIFY(A,i). Illustrate the operation of MAX-HEAPIFY(A,3) on the array A={27, 17, 3, 16, 13, 10, 1, 5, 7, 12, 4, 8, 9, 0} (2.5)

b) Write the Insertion Sort algorithm. Analyze the best case and worst case time complexity of this algorithm. (2.5)

Q6 a) Find an optimal solution to the following knapsack instance: (2.5)
Number of items=n=8, Knapsack Capacity=M=17. Profit=P={10, 15, 8, 7, 3, 15, 8, 27} and weight=W= {5, 4, 3, 7, 2, 3, 2, 6}. Write a suitable algorithm to solve this problem.

b) Describe the PARTITION() algorithm of QUICK-SORT() step by step how you would get the pass1 result by taking last element as pivot on the following data. (2.5)

2, 5, 7, 5, 9, 3, 8, 6

Q7 a) Write pseudocode for the procedure HEAP-EXTRACT_MIN. Explain with the given min-heap level order traversal A={10, 20, 15, 22, 30, 18, 17, 40, 28}. (2.5)

b) Show that there are at most $|n/2^{h+1}|$ nodes of height h in any n-element heap. (2.5)

==========