Nirma University

Institute of Technology

Semester End Examination (IR), May - 2014

B. Tech. in Computer Engineering, Semester-VI

2CE339 Analysis and Design of Algorithms

Roll / Exam No. _____ Supervisor's initial with date _____

Time: 3 Hours Max Marks: 100

Instructions:
1. Attempt all questions.
2. Figures to right indicate full marks.
3. Use section-wise separate answer book.
4. Draw neat sketches wherever necessary.

SECTION –I

Q-1 Answer the following: **[18]**

[A] Give an optimized algorithm to find out k^{th} smallest element from a given array. (For example, if your array is 8, 5, 13,-9, 20, 16 and k is 3 then your algorithm should display 8.) [5]

[B] What value is returned by the following function? Express your answer as a function of n. Give the worst-case running time using the Big Oh notation. [5]

```
function mystery(n)
    r:= 0
    for i:= 1 to n-1 do
        for j:= i+1 to n do
            for k:= 1 to j do
                r:= r+1
    return(r)
```

[C] Derive the time complexity for an algorithm to find n^{th} Fibonacci number. [5]

[D] "Worst Case Analysis is better than best case analysis". Do you agree with this statement? Justify your answer. [3]

Q-2 Answer the following: **[16]**

[A] For a given problem P, two algorithms A and B have respective complexities $T_1(n)$ and $T_2(n)$ in terms of size n, where [5]

$T_1(n) = 4n^5+3n$

$T_2(n) = 2500n^3+4n$

Find the range of n, the size of an instance of the given problem, for which A is more efficient than B.

OR

[A] Compare the algorithms indicated by following recurrences and state your conclusion with respect to time complexity. [5]

(a) $T(n)=4T(n/2)+n$
(b) $T(n)=4T(n/2)+n^2$
(c) $T(n)=4T(n/2)+n^3$

[B] Guess the solution to the recurrence (Use intelligent guesswork): [5]

$$t_n = 0 \quad \text{if } n = 0$$
$$= 3t_{n-1} + 15 \quad \text{otherwise}$$

[C] Explain following terms with appropriate example : [6]
a) Limit rule
b) Maximum rule
c) Smoothness rule

Q-3 Answer the following: **[16]**

[A] Calculate minimum number of scalar multiplications required for computing M^6 where dimensions of matrix M are 4x4. [3]

[B] A bottleneck spanning tree T of an undirected graph G is a spanning tree of G whose largest edge weight is minimum over all spanning trees of G. We say that the value of the bottleneck spanning tree is the weight of the maximum-weight edge in T. [6]

(a)Argue that a minimum spanning tree is a bottleneck spanning tree.
(b)Give a linear-time algorithm that given a graph G and an integer b, determines whether the value of the bottleneck spanning tree is at most b.

OR

[B] Solve the following : [6]
The input to the problem is a sequence S of integers (not necessarily positive). The problem is to find the consecutive subsequence of S with maximum sum. "Consecutive" means that you are not allowed to skip numbers. For example if the input was 12, −14, 1, 23, −6, 22, −31, 13 the output would be 1, 23, −6, 22. Give a linear time algorithm for this problem.

[C] Compare the greedy and dynamic programming solution for the 0/1 knapsack problem with following parameters: [7]

W = 38, n=4, v = (11,20,15,31) and w=(2,9,18,15)

Section II

Q-4 Answer the following: **[16]**

[A] Describe the principle of optimality in Dynamic Programming with suitable example. [4]

[B] Explain the concept of binomial trees and operations on them with appropriate example. [6]

[C] Distinguish between divide and conquer and Dynamic Programming approach for algorithm design. [6]

OR

[C] Your task is to optimize a code. Will you go only for sequential optimization? Do you think parallelizing the code can further optimize it? Give your thoughts and discuss some of the issues which may be faced [6]

during such enhancement.

Q-5 Answer the following: **[16]**

[A] Discuss benefits and shortcomings of greedy algorithms. [5]

[B] Show the construction of Min Heap for the following set of data : 56, 50, 57, 55, 44, -52, 54, 66, 34 [5]

[C] Solve the following recurrence: [6]
T (n) = 2T (n/2) + nlgn, where n is power of 2 and n >= 2.

Q-6 Answer the following: **[18]**

[A] Consider the following set of activities (I), their starting times (Si) and finishing times (Fi) are : [8]

I = <i1,i2,i3,i4,i5,i6,i7,i8,i9>
Si = <1,3,0,5,3,5,6,8,8>
Fi = <4,5,6,7,8,9,10,11,12>

(For example, in the above set, the activity i2 starts at time 3 and finishes at time 5). Moreover, the activities are arranged in the monotonically increasing order of finishing times.

Assuming that only one activity can be scheduled at a time (i.e., no activity can start before the finishing time of current activity), find the set of activities that will be selected or scheduled by a greedy activity selector whereas the job of a greedy activity selector is to schedule several computing activities that require exclusive use of a common resource i.e., it selects the maximum number of mutually exclusive activities.

OR

[A] Compute the time complexity to calculate a^n, where n>0 and a is a positive integer with m digits. [8]

[B] Write a program which accepts a string as input and returns 1 if the number of characters in the string is even else returns 0. Design an algorithm to solve the above problem and analyze it in terms of the resource required. [6]

[C] Distinguish between apriori and posteriori approach for analysis of algorithms. [4]