Roll No.: ____________________

Amrita Vishwa Vidyapeetham
B.Tech. Second Assessment – April 2017
Fourth Semester
(Computer Science and Engineering)

15CSE211 Design and Analysis of Algorithms

Time: Two hours Maximum: 50 Marks

Answer all questions

Part A: **5*2 = 10 Marks**

1. The recurrence $T(n) = 7T(n/2) + n^2$ describes the running time of an algorithm A. A competing algorithm B has a running time of $T(n) = aT(n/4) + n^2$. What is the largest integer value for 'a' such that B is asymptotically faster than A?
2. Consider the matrix chain multiplication problem where the goal is to optimally parenthesize n matrices. Consider $A_1..A_{n-1}$ as the matrices, each A_i with dimensions d_i X d_{i+1}. Let $N_{i,j}$ be the minimum number of scalar multiplications to multiply the matrices A_i through A_j. Give the recurrence equation to compute $N_{i,j}$.
3. Consider the following recursive algorithm.

```
public static int f(int n, int r) {
   if(r == 0 || r >= n) return 1;
   else return f(n - 1, r - 1) + f(n - 1, r);
}
```

   Using dynamic programming, convert this into a method that takes $O(n \cdot r)$ time.

4. Consider the problem of finding the longest increasing subsequence from a set of integers. Would you use a top-down approach, or a bottom-up approach? Justify your answer.
5. There are eight identical-looking coins; one of these coins is counterfeit and is known to be lighter than the genuine coins. What is the minimum number of weighings needed to identify the fake coin with a two-pan balance scale without weights?

Part B: **5*8 = 40 Marks**

1. Answer the following questions.
   (a) Draw the recursion tree for $T(n) = 4T(n/2) + cn$, where c is a constant, and provide a tight asymptotic bound on its solution. [4]
   (b) Solve the following recurrence relations using Master Method. [4]
      i. $T(n) = 2T(n/2) + n^3$
      ii. $T(n) = 16T(n/4) + n^2$.

2. Answer the following.
   (a) Consider an array of distinct numbers sorted in increasing order. The array has been rotated (anti-clockwise) *k* number of times. Given such an array, your goal is to find the value of *k* i.e., the number of times it has been rotated. Give an efficient logarithmic time algorithm to solve this problem. [5]
   (b) Suppose your job is to sell advertising space on a web page. At most half the screen can be filled with ads, and for each potential ad buyer we can associate a pair of numbers (*f*,*p*) where *f* is the fraction of a screen the buyer's ad would take and p is the price he's willing to pay for his ad. Your job is to choose a subset of the ads that maximizes your total profit. Ads cannot be resized; you must take them at the given size or not take them at all. Would this problem be more naturally modeled as a fractional knapsack problem, or a 0-1 knapsack problem? Explain your answer. [3]

3. You are given a set of strings and your goal is to find the longest common prefix of these strings. For e.g input: {"gradschool", "grads", "graduation", "grade"}; output is "grad". Give an efficient algorithm to solve this problem. Explain your algorithm details, its correctness and complexity. [8]

4. Several coins are placed in cells of an n × m board, no more than one coin per cell (cells may be empty also). A robot, located in the upper left cell of the board, needs to collect as many of the coins as possible and bring them to the bottom right cell. On each step, the robot can move either one cell to the right or one cell down from its current location. When the robot visits a cell with a coin, it always picks up that coin. Design an algorithm to find the maximum number of coins the robot can collect and a path it needs to follow to do this. Explain your algorithm details, its correctness and complexity. [8]

5. Consider the subset-sum problem: find a subset of a given set A = {a1 , . . . , an} of n positive integers whose sum is equal to a given positive integer d. Give a backtracking approach to solve this problem. Explain your algorithm using examples. What is the complexity of your algorithm? [8]