ROLL NO. _______________

Code: CT42 **Subject: DESIGN AND ANALYSIS OF ALGORITHMS**

ALCCS

Time: 3 Hours **December - 2017** **Max. Marks: 100**

PLEASE WRITE YOUR ROLL NO. AT THE SPACE PROVIDED ON EACH PAGE IMMEDIATELY AFTER RECEIVING THE QUESTION PAPER.

NOTE:

- **Question 1 is compulsory and carries 28 marks. Answer any FOUR questions from the rest. Marks are indicated against each question.**
- **Parts of a question should be answered at the same place.**

Q.1 a. Explain how one can generate a Huffman code without an explicit generation of a Huffman coding tree.

b. Explain “Divide and Conquer Technique”.

c. Find a solution to the 4-Queens problem using back-tracking techniques.

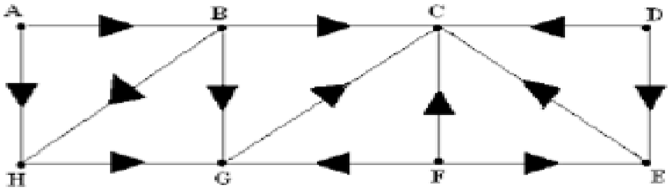
d. Using an example, explain dynamic programming.

e. Suppose that the root of the Red-Black tree is red. If we make it black, does the tree remain a Red Black tree?

f. Show that the worst case complexity for simple text search (Naive string matching) to find the first occurrence of a pattern of length m on a text of length n is θ(n-m+1)(m-1).

g. Define: “topological sort” of a directed acyclic graph. What is the time complexity of topological sort? **(4×7)**

Q.2 a. Write an algorithm to sort a Directed Acrylic Graph (DAG) topologically. Show ordering of vertices produced by topological sort when it is run on the following DAG. **(12)**



b. Give an efficient algorithm that determines whether or not a given directed graph G = (V, E) contains a cycle. Discuss its time complexity. **(6)**

Q.3 a. Write down the algorithm for Radix Sort. Illustrate your algorithm on the following list of English words: **(9)**
COW, DOG, SEA, RUG, ROW, MOB, BOX, TAB, BAR, EAR, TAR, DIG, BIG, TEA, NOW, FOX

b. Draw a graph with 6 vertices that has unique ordering of vertices when topologically sorted. **(2)**

c. Let G be an undirected connected graph. Give an efficient algorithm to compute the second best minimum spanning tree of G. **(7)**

Q.4 a. Define AVL tree. Construct an AVL tree for the list 3, 6, 5, 1, 2, 4. **(9)**

b. Give a linear time in-place algorithm to rearrange an array of n keys so that all the even-valued keys precede all the odd-valued keys. Show that your algorithm runs in linear time. **(9)**

Q.5 a. What do you understand by "Closest Pair of Points Problem"? How can you compute closet pair of points in (i) O (n^2) time (ii) O (n log n) time. **(9)**

b. Suppose that in 0-1 knapsack problem the order of items when sorted on increasing value is the same as their order when sorted by decreasing weight. Give an efficient algorithm to find an optimal solution to the problem. **(9)**

Q.6 a. Given the characters S <a, b, c, d, e, f> with the following probability P = <29, 25, 20, 12, 05, 09>. Build a binary tree using greedy Huffman algorithm. **(8)**

b. Write down Knuth Morris Pratt algorithm for string matching. Compute the prefix function for the pattern b a c b a b a b a a b c b a b **(10)**

Q.7 a. Prove that Hamiltonian cycle problem is polynomial time verifiable. **(9)**

b. When a decision problem D1 is said to be **(6)**
(i) polynomially reducible?
(ii) NP-complete?

c. If A is NP-complete then A is a member of P if and only if P = NP. **(3)**