ROLL NO. _____

Code: CT42 **Subject: DESIGN & ANALYSIS OF ALGORITHM**

ALCCS

Time: 3 Hours **JUNE 2015** **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. Find a solution to the 4-Queens problem using back-tracking techniques.

b. Give an Adjacency List representation for a complete binary tree on 15 vertices. Give an equivalent Adjacency Matrix representation. Assume that vertices are numbered from 1 to 15 as in a binary heap.

c. Illustrate the operation of Counting Sort on the array A=(6,0,2,0,13,4,6,1,3,2)

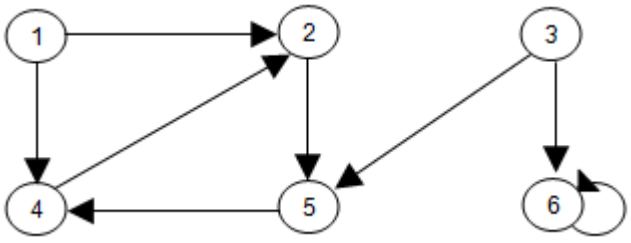
d. Show all legal B-trees of minimum degree 2 that represent (1,2,3,4,5)

e. Give a simple example of a directed graph with negative weight edges for which Dijkstra's algorithm produces in correct answers.

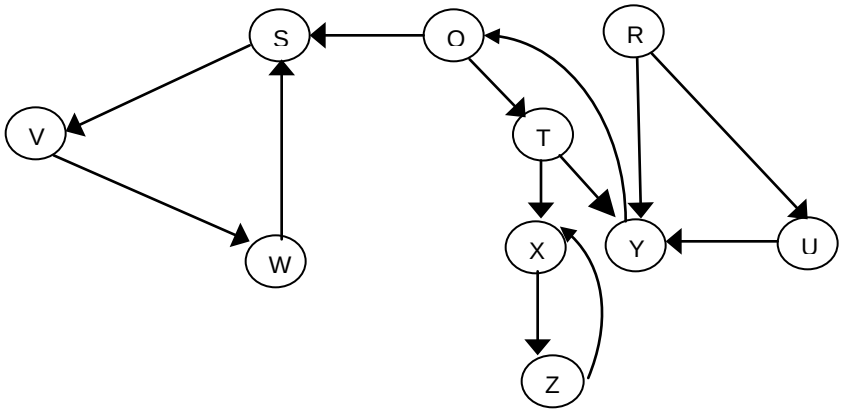
f. State Max Flow – Min Cut Theorem.

g. Show the comparisons the naive string matcher makes for the pattern P=0001 in the text T = 000010001010001 **(7×4)**

Q.2 a. What are the different ways of representing a graph in the memory of a computer? Represent the following graph using any three ways. **(9)**



b. Write an algorithm for Depth First Search on a directed graph. Show how your algorithm will work on the following graph. Your algorithm should consider the vertices in alphabetical order and assume that each adjacency list is ordered alphabetically. **(9)**



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. Define Max-Heap and Min-Heap property. Write an algorithm for creating a max-Heap when the input is in the form of an array of integer numbers. Illustrate the operation of your algorithm on the array A= (27, 17, 3, 16, 13, 10, 1, 5, 7, 12, 4, 8, 9, 0) **(9)**

Q.4 a. What do you mean by hashing? Explain any five popular hash functions. Draw the 11-item hash table resulting from hashing the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, 5 using the has function h(i) = (2i+5) mod 11 **(9)**

b. Taking a suitable example explain how disjoint sets are represented using linked lists. Show the data structure that results and the answers returned by the Find_Set operations in the following programme. Use the linked-lit representation with the weighted union heuristics. Assume that if the sets containing X_i and X_j have the same size, then Union (X_i, X_j) appends X_j's list on to X_i's list. **(9)**

for i = 1 to 16
    Make –Set (X_i)
for i = 1 to 15 by 2
    Union (X_i, X_{i+1})
for i = 1 to 13 by 4
    Union (X_i, X_i+2)
Union (X_1, X_5)
Union (X_{11}, X_{13})
Union (X_1, X_{10})
Find-Set (X_2)
Find_Set (X_9)

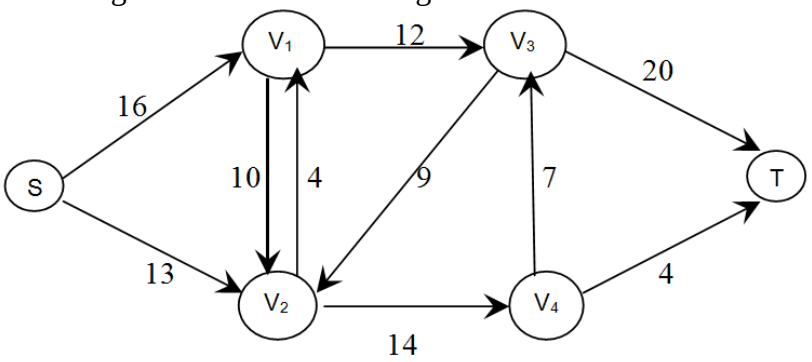
Q.5 a. What do you understand by "Closest Pair of Points Problem"? How can you compute closet pair of points in (i) O (n^2) time (ii) O (n log n) time. **(6)**

b. Suppose that a graph G has a minimum spanning tree already computed. How quickly can we update the minimum spanning tree if we add a new vertex and incident edges of G. **(6)**

c. Give an algorithm to compute the second best minimum spanning tree of a graph G.**(6)**

Q.6 a. Write down the recursive formula that generates the optimal sub-structure of Longest Common Subsequence (LCS) problem. Determine an LCS of (b, a, a, b, a, b, a, b) and (a, b, a, b, b, a, b, b, a) **(9)**

b. Write down Ford-Fulkerson method to find the maximum flow in a flow-network. Execute the algorithm on the following flow-network. **(9)**



Q.7 a. Write down Knuith-Morris-Pratt(KMP) algorithm for string matching. Compute the prefix function for the pattern: ababbabbabbababbabb **(12)**

b. Prove that (i) if any NP-complete problem is polynomial-time solvable, then P=NP (ii) if any problem in NP is not polynomial time solvable, then no NP-complete problem is polynomial time solvable. **(6)**