Code: C-06 / T-06 **Subject: DATA STRUCTURES & ALGORITHM DESIGN**
Time: 3 Hours **June 2006** **Max. Marks: 100**

NOTE: There are 9 Questions in all.

- **Question 1 is compulsory and carries 20 marks. Answer to Q. 1. must be written in the space provided for it in the answer book supplied and nowhere else.**
- **Out of the remaining EIGHT Questions answer any FIVE Questions. Each question carries 16 marks.**
- **Any required data not explicitly given, may be suitably assumed and stated.**

Q.1 Choose the correct or best alternative in the following: (2x10)

a. The number of unused pointers in a complete binary tree of depth 5 is

(A) 4 **(B)** 8
(C) 16 **(D)** 32

b. The running time for creating a heap of size n is

(A) O (n) **(B)** O (log n)
(C) O (n log n) **(D)** O (n^2)

c. A complete binary tree with the property that the value of each node is at least as large as the values of its children is known as

(A) Balanced Tree **(B)** AVL Tree
(C) Heap **(D)** BST

d. What would be returned by the following recursive function after we call test (0, 3)

```
int test (int a, int b)
{
   if (a==b) return (1);
   else if (a>b) return(0);
   else return (a+test(a+1, b));

}
```

(A) 1 **(B)** 2
(C) 3 **(D)** 4

e. The extra key inserted at the end of the array is called a

(A) End Key **(B)** Stop Key
(C) Sentinel **(D)** Transposition

f. Which of the following operations is performed more efficiently by doubly linked list than by singly

linked list

(A) Deleting a node whose location is given.
(B) Searching of an unsorted list for a given item.
(C) Inserting a new node after node whose location is given.
(D) Traversing the list to process each node.

g. One can determine whether a Binary tree is a Binary Search Tree by traversing it in

(A) Preorder **(B)** Inorder
(C) Postorder **(D)** Any of the three orders

h. The spanning tree of connected graph with 10 vertices contains

(A) 9 edges **(B)** 11 edges
(C) 10 edges **(D)** 9 vertices

i. A sorted file contains 16 items. Using binary search, the maximum number of comparisons to search for an item in this file is

(A) 15 **(B)** 8
(C) 1 **(D)** 4

j. One can determine whether an infix expression has balanced parenthesis or not by using

(A) Array **(B)** Queue
(C) Stack **(D)** Tree

Answer any FIVE Questions out of EIGHT Questions.
Each question carries 16 marks.

Q.2 a. How can we modify almost any algorithm to have a good best case running time? **(4)**

b. Evaluate the following postfix-expression using stack. Show the content of the stack in each step.
6 2 3 + - 3 8 2 / + * 2 $ 3 + **(6)**

c. Sketch an algorithm to find the minimum and the maximum elements from a sequence of n elements. Your algorithm should not take more than (3n/2)-2 number of comparisons where n is a power of 2. **(6)**

Q.3 a. Write an O(1) algorithm to delete a node p in a singly linked list. Can we use this algorithm to delete every node? Justify. **(7)**

b. Suggest an efficient sorting algorithm to sort an array of n large records, while minimizing number of exchanges. **(2)**

c. Write an algorithm that will split a circularly linked list into two circularly linked lists. **(7)**

Q.4 a. Let A[n] be an array of n numbers. Design algorithms to perform the following operations:

Add (i, y): Add the value y to the i^{th} number in the array
Partial-sum(i) : returns the sum of the first i numbers in the array **(4)**

b. Calculate the efficiency of Quick sort. **(4)**

c. Formulate an algorithm to perform insertion sort on a linked list. **(8)**

Q.5 a. What is hashing? Explain why we require the table size to be a prime number in double hashing? **(5)**

b. The following values are to be stored in a Hash-Table:
25, 42, 96, 101, 102, 162, 197, 201

Use the Division method of Hashing with a table size of 11. Use the sequential method for resolving collision. **(6)**

c. Rewrite your solution to part **(b)** using rehashing as the method of collision resolution. Use [(key+3) mod table-size] as rehash function. **(5)**

Q.6 a. How many AVL trees of 7 nodes with keys 1, 2, 3, 4, 5, 6, 7 are there? Explain your answer. **(6)**

b. Give an AVL tree for which the deletion of a node requires two double rotations. Draw the tree and explain why two rotations are needed? **(10)**

Q.7 a. The ordinary linked representation of a binary tree with n nodes has exactly (n+1) unused (nil) pointers. Why? Explain with an example. **(4)**

b. A funny tree is a binary tree such that, for each of its nodes x, the number of nodes in each sub tree of x is at most 2/3 the number of nodes in the tree rooted at x. Draw the tallest funny tree of 5 nodes. **(7)**

c. T1 and T2 are two arbitrary binary trees each having n unlabelled nodes. Show that it is sufficient to apply at most 2 (n-1) single rotations to T1 so that it becomes equal to

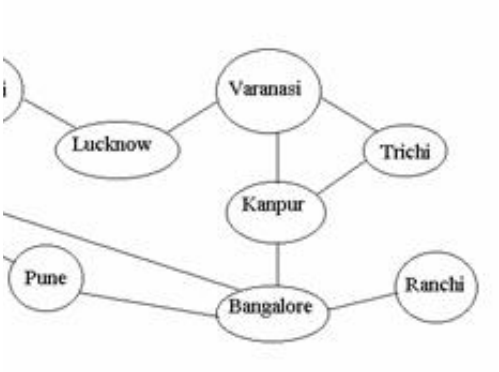
T2. **(5)**

Q.8 a. Suppose that a Graph has a minimum-spanning tree already computed. How quickly can the minimum spanning tree be updated if a new vertex and incident edges are added to G? **(4)**

b. Describe the Dijkstra's algorithm for finding a shortest path in a given graph. **(8)**

c. Explain the difference between depth first and breadth first traversing techniques of a graph. **(4)**

Q.9 a. Consider the following undirected graph and answer the following questions.



Assume that the edges are ordered alphabetically (i.e. when facing with alternatives, choose the edges in alphabetical order)

(i) List the nodes (cities) of the graph by depth first search starting from **Varanasi.**
(ii) List the nodes (cities) of the graph by breadth first search starting from **Calcutta**. **(8)**

b. Consider the following function
F(int n, int m)
{if n < = 0 OR m<= 0 then return 1 else return (F(n-1, m) + F(n, m-1));}

Now answer the following questions assuming that n and m are positive integers.
(i) What is the value of F(n, 2), F(5, m), F(3, 2), F(n, m)?
(ii) How many recursive calls are made to the function F, including the original call when evaluating F(n, m)? **(4)**

c. Write a recursive function that has one parameter which is an integer value called x. The function prints x asterisks followed by x exclamation points.

Do not use any loops. Do not use any variables other than x. **(4)**