FACULTY OF ENGINEERING

B.E. 2/4 (CSE) I-Semester **(Old)** Examination, June / July 2012

Subject : **Data Structures**

Time : 3 Hours

Max. Marks: 75

Note: Answer **all** questions of Part - A and answer any **five** questions from Part-B.

PART – A (25 Marks)

1. Define the term Data structure and state the difference between linear data structure and non-linear data structure. (2)
2. Define minimum spanning tree. How many edges exists for a tree of n nodes in it? (2)
3. Define transitive closure. Explain it use in graphs. (3)
4. Show the results of running shell sort on the input 9, 8, 7, 6, 5, 4, 3, 2, 1 using the increments {1, 3, 7}. (3)
5. Define priority queue. Name any two applications in computer science where priority queue is used. (3)
6. What is quadratic probing? (2)
7. State the salient differences between AVL tree and splay tree. (2)
8. Define threaded binary tree. (2)
9. How do you evaluate a post fix expression? Discuss with an example. (3)
10. Define Isomorphic property of tree. What is the running time to decide whether two trees T_1 and T_2 are isomorphic or not. (3)

PART – B (5x10=50 Marks)

11. Given two sorted listed L_1 and L_2 , write a procedure to compute $L_1 \cup L_2$ using only the basic list operations. (10)

12. Write a non-recursive function to reverse a singly linked list in O(N) time. (10)

13.(a) Write a non recursive function to insert into an AVL tree.
   (b) Define B-tree. Write a routine to insert an element into B-tree. (10)

14.(a) Write an algorithm to implement Krushkal's algorithm. (5)
   (b) Find the minimum spanning tree for the following graph: (5)
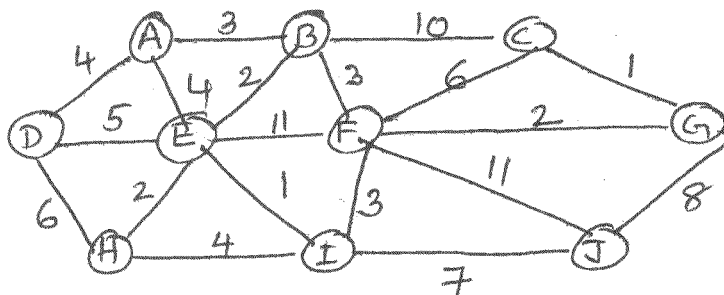
15.(a) Determine running time of merge sort for
   (i)Sorted input   (ii) reverse order input   (iii) random input (5)
   (b) Insert the following keys into hash table of size 5 use hash function K%5 and linear probing to resolve collisions 21, 35, 22, 37, 27, 38. (5)

16. Discuss extendible hashing with examples. (10)

17. Write short notes on the following : (4+2+4)
   (a) Differences between splay tree and AVL tree
   (b) Simulating pointers