MCS-021

01615

MCA (Revised)

Term-End Examination

June, 2009

MCS-021 : DATA AND FILE STRUCTURES

Time : 3 hours *Maximum Marks : 100*

(Weightage 75%)

Note : *Question number **1** is **Compulsory**. Attempt **any three** questions from the rest. All algorithms should be written nearer to C language.*

1. (a) Define the term "Complexity". For what types of applications is Time Complexity Critical and for what types of applications is Space Complexity Critical. **10**

(b) Write an algorithm for 2-way merge sort. Sort the following numbers and show all intermediate steps : **10**

7, 18, 39, 5, 72, 115, 13, 44, 56

(c) How AA Trees are better than Red-Black trees? Construct an AA-tree using the following nodes. Show all intermediate steps and balancing of tree. **10**

8, 15, 25, 89, 5, 70, 32, 45, 33, 12, 17, 94, 112, 13

(d) Write an algorithm to implement the following functions of a dequeue. Also, give an example to show the working of these functions. **10**

(i) Insert Element

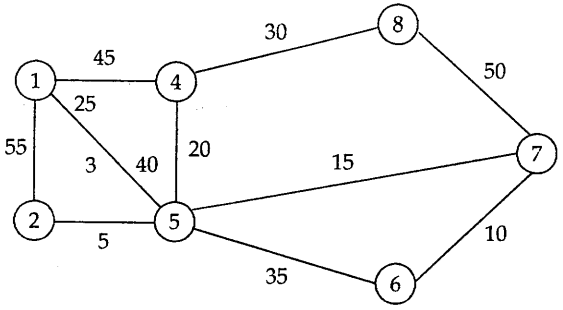
(ii) Create DEQUEUE

(iii) Delete Element

2. (a) Write an algorithm for evaluating postfix expression. Show the working of your algorithm on the following expression. **10**

[10, 8, +, 12, 9, 5, /, +, *]

(b) Find the minimum Cost Spanning Tree of the following graph. **10**



3. (a) Explain a circular linked list. Write an algorithm to insert and delete an element in circular linked list. **10**

(b) Write an algorithm for multiplication of two sparse matrices. **10**

4. (a) What is cylinder - surface indexing ? Explain it with an example. Also, write its merits and demerits. **7**

(b) Define Abstract Data Type, and give two examples of it. **3**

(c) Construct a binary tree using the following pre-order and in-order traversals : **10**

Pre-order : A B G H M C D E F

In-order : B H M G A D F E C

5. (a) Construct a B-Tree for the following data. Show all intermediate steps during the process. **10**

D H K Z B P O Q E A S W T C L N Y M

(b) Explain the following using an example : **10**

(i) Breadth First Search

(ii) AVL Tree

- o O o -