Code: 17F00201

MCA II Semester Regular & Supplementary Examinations May/June 2019

DATA STRUCTURES

(For students admitted in 2017 & 2018 only)

Time: 3 hours Max. Marks: 60

Answer all the questions

*****

1 (a) Justify the statement 'Pointers can be Dangerous' with an illustration.

(b) A 'C' function for printing out a matrix is as follows:

```
Void print matrix(int matrix [ ] [MAX_SIZE], int rows, int cols)
{int i, j;
for (i = 0;i<rows;i++)
for(i = 0;i<cols;i++)
{printf("%d", matrix[i] [j];
printf("\n"]);
 }
}
```

Determine worst-case complexity for the above 'C' function.

OR

2 (a) Given 'n', a positive integer, determine if 'n' is the sum of its divisors, i.e if 'n' is the sum of all 't' such that 1≤t <n and 't' divides 'n'.

(b) Explain 'clocking' in performance measurement with functions supported in 'C'.

3 (a) Explain Dynamically allocated one dimensional array with an example.

(b) Write a 'C' program for demonstrating the various stack operations, including cases of overflow and underflow of stacks.

OR

4 (a) Define stack. Implement 'Push' and 'Pop' functions for stack using arrays.

(b) Explain with a suitable example, how circular queue is implemented using dynamically allocated arrays.

5 (a) Give the node structure to create a singly linked list of integers and write functions to perform the following:

(i) Create a list

(ii) Assume the list contains 3 nodes with data 10, 20, 30. Insert a node with data 40 at the end of the list.

(iii) Insert a node with data 50 between the nodes having data values 10 and 20.

(iv) Display the singly linked list.

(b) Write the node structure for linked list representation of a polynomial.

OR

6 (a) List out the difference between the Doubly linked list and singly linked list.
Illustrate with example the following operations on a Doubly linked list.

(i) Inserting a node at the beginning.

(ii) Inserting at the intermediate position.

(iii) Deletion of a node with a given value.

(b) Explain 'Spare matrix representation' with an example.

Contd. in page 2

Code: 17F00201

7 (a) Define binary trees. Explain the following with examples:
(i) Complete binary tree.
(ii) Skewed binary tree.
(iii) Almost complete binary tree.
(iv) Height of a binary tree.
(b) Explain 'Spanning tree' with an example.

OR

8 (a) What is the advantage of threaded binary tree over binary tree? Explain the construction of threaded binary tree for 10, 20, 30, 40 and 50.
(b) Define graph. Outline the difference between graphs and trees with examples.

9 (a) What is a heap? Write an algorithm to implement heap sort.
(b) Write a 'C' program to accept an array of 'n' integer elements and searches for the desired element using binary search.

OR

10 (a) Write an algorithm for quick sort. Trace the algorithm for the data: 45, 26, 77, 14, 68, 61, 97, 39.
(b) Explain Fibonacci search with an example.

*****