Reg. No.									

MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent unit of MAHE, Manipal)

VI SEMESTER B.TECH (ELECTRICAL & ELECTRONICS ENGINEERING)

MAKE UP EXAMINATIONS, JUNE 2018

SUBJECT: DATA STRUCTURES & ALGORITHM [ELE 4018]

REVISED CREDIT SYSTEM

Time: 3 Hours **Date: 22nd June 2018** **Max. Marks: 50**

Instructions to Candidates:

- Answer **ALL** the questions.
- Missing data may be suitably assumed.

1A. Calculate the run-time efficiency of the following program segments. ***(04)***

Code-1	**Code-2**

int j
for (j = 1000; j >= 1; j = j / 2) {
Application code}
	int i, j;
	for (i = 0; i < 10; i++) {
	for (j = 0; j < i; j ++) {
	Application code }}

1B. A square matrix is called symmetric if for all values of **i** and **j**, $a[i][j] = a[j][i]$**.** Write a pseudo-code, with detailed comments, which verifies whether a given 4 x 4 matrix is symmetric. ***(02)***

1C. A lower-triangular matrix is shown below. Derive an expression for the address of a_{ij}th element if the address of the first element of the matrix is **M**. Assume that the elements are stored in memory as **row-major**.

$$A = \begin{bmatrix} a_{1,1} & 0 & \dots & 0 & 0 \\ a_{2,1} & a_{2,2} & \dots & 0 & 0 \\ a_{3,1} & a_{3,2} & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ a_{n,1} & a_{n,2} & \dots & a_{n,n-1} & a_{n,n} \end{bmatrix}$$

(04)

2A. Let there be a single linked list whose address of the first node is stored in a pointer called **LIST**. Write a pseudo-code, with detailed comments, which has 2 subroutines to perform the following operations:

a) To find the number of times (say, **NUM**) a given item (say, **ITEM**) occurs in the list.

b) To find the number of nonzero elements (say, **NONZERO**) in LIST. ***(04)***

2B. Write a pseudo-code, with detailed comments, to delete a node from a double linked list if the key (say, **KEY**) is found anywhere in the list (which includes the first and the last nodes). Assume that the elements of the list are unique and the address of the 1st node is stored in a pointer called **HEADER**. ***(04)***

2C. Represent the arithmetic expression $P = ((A + ((B\hat{}C) - D)) * (E - (A/C)))$ in prefix and postfix notations ***(02)***

3A. Postfix representation of an arithmetic expression is $\mathbf{ABC} * \mathbf{D} /+$ **.** Show detailed stack operation, in tabular form, involved in the evaluation of the expression.
Take A = 4, B = 6, C = 8, D = 12 ***(03)***

3B. Write separate sub-routines, with detailed comments to enqueue and dequeue **DATA** from a queue which is implemented using a single-linked list. Given that, **REAR** points to the enqueued end and **FRONT** points to the dequeued end. ***(04)***

3C. A circular queue, **Q**, is implemented by an array of specified size (say, **SIZE**). **FRONT** and **REAR** are index numbers from deletion and insertion ends of the circular queue respectively. Write a pseudo-code, with detailed comments to insert **DATA**. ***(03)***

4A. For the tree shown in Fig. 4A answer the following questions

a. What is the depth of the tree?	d. What is the level of E?
b. Which nodes are children of node B?	e. Which nodes are siblings of node H?
c. Which node is the parent of node F?	f. Which nodes are the siblings of node D?

(03)

4B. A Binary Search Tree (BST), containing integer data, is implemented using linked list. Write a pseudo-code, with detailed comments, to search for an integer, (say **VAL**). If VAL is found then print "VAL found" otherwise print "VAL not found". The address of the root node is stored in a pointer called **ROOT**. ***(03)***

4C. A max heap tree is created with the numbers **33, 42, 67, 23, 44, 49, 74**. Show the steps involved diagrammatically. Once the heap tree is created show the steps involved in inserting **99** in the tree diagrammatically. ***(04)***

5A. Explain the breadth first search (BFS) traversal for the graph shown in Fig. 5A below. Also draw the spanning tree. ***(06)***

5B. Write down the adjacency matrix of the digraph shown in Fig. 5B below. Draw the linked list representation of the graph. Write down the adjacency matrix if a new node, v8, is inserted between nodes v1 and v4 such that the edges are directed from v4 to v8 and v8 to v1. ***(04)***

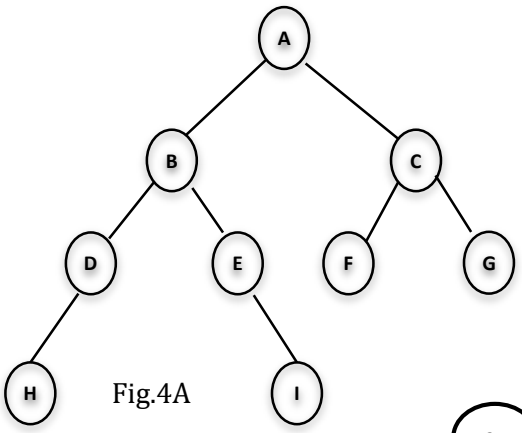

Fig.4A

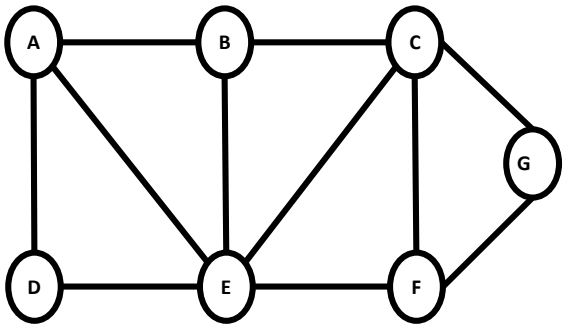

Fig.5A

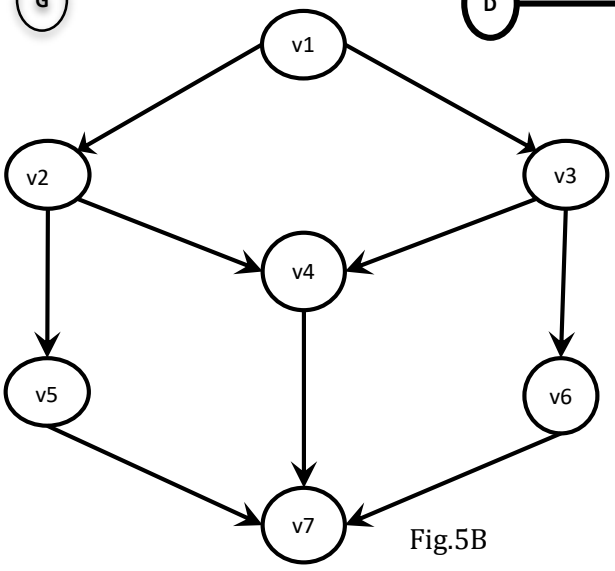

Fig.5B