Total No. of Questions—8]   [Total No. of Printed Pages—4

<table>
<tr><td>Seat
No.</td><td></td></tr>
</table>

[4657]-545

S.E. (E & TC/Electronics) (First Semester) EXAMINATION, 2014

DATA STRUCTURES AND ALGORITHMS

(2012 PATTERN)

Time : Two Hours   **Maximum Marks : 50**

N.B. **:—** (*i*) Answer Q. No. **1** *or* Q. No. **2** and Q. No. **3** *or* Q. No. **4** and Q. No. **5** *or* Q. No. **6** and Q. No. **7** *or* Q. No. **8**.

  (*ii*) Assume suitable data if necessary.

  (*iii*) Write output of programs if necessary.

  (*iv*) Draw neat diagrams wherever necessary.

1. (*a*) Define recursion with example in detail.   [4]

  (*b*) Write a function in 'C' to sort numbers using bubble sort.   [4]

  (*c*) Differentiate between static memory and dynamic memory allocation.   [4]

Or

2. (*a*) Write algorithm to add two polynomials using array of structures.   [4]

P.T.O.

(*b*) Define pointers and write function 'swap' to swap two numbers using pointers. [4]

(*c*) Explain bitwise operators with example in detail. [4]

3. (*a*) Differentiate between SLL and DLL. [4]

(*b*) Write function PUSH to implement stack using array. [4]

(*c*) Name types of queues. Explain any *one* in detail. [4]

Or

4. (*a*) Write a function in 'C' to delete a node in SLL. [4]

(*b*) Explain operation insert in linear queue. [4]

(*c*) Convert the following infix expression to postfix using stack : [4]

a + b*(c/d $ a)/b.

5. (*a*) Define BST. Create BST for the following numbers : [5]

56, 34, 89, 11, 45, 67, 6, 78.

Show preorder traversal.

(*b*) Explain threaded binary tree with an example. [4]

(*c*) Define the following terms with example : [4]

(*i*) Strictly binary tree

(*ii*) Completely binary tree.

Or

6. (*a*) Write a function in 'C' to search a number in BST. [5]

(*b*) For the following numbers create AVL tree : [4]

6 5 4 3 2 1.

(*c*) Explain expression trees with *one* example. [4]

7. (*a*) Using Dijkstra's algorithm find out shortest path four source node ⓐ for the given graph : [5]
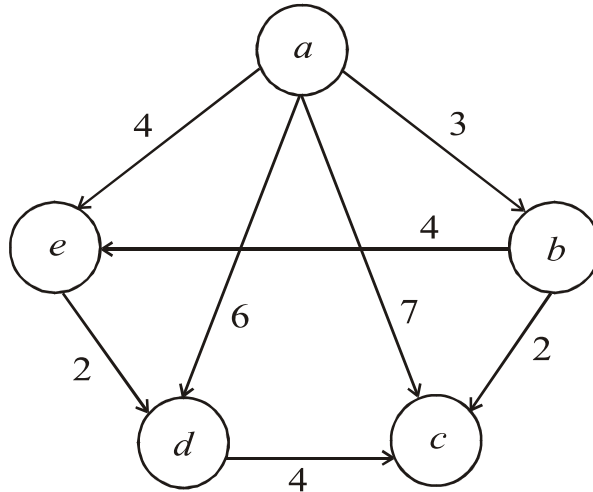

Fig. 1

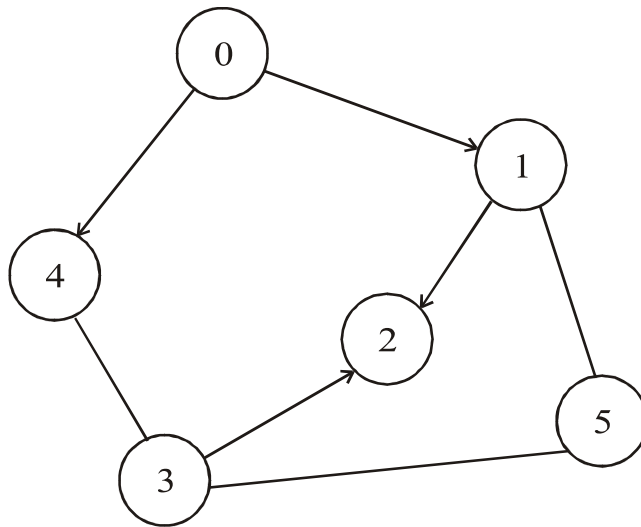(*b*) Represent the given graph using adjacency matrix : [4]

Fig. 2

(*c*) Explain Kruskal's algorithm with an example. [4]

Or

8. (*a*) Explain graph traversal methods with suitable example. [5]

(*b*) Explain Prim's algorithm with suitable example. [4]

(*c*) Define with example : [4]

(*i*) Connected graph

(*ii*) Path.