44 (Sem–2) 2·1 (N/O)

2019

DATA STRUCTURE AND ALGORITHMS

Paper : 2·1

(New Syllabus)

Full Marks : 80

Time : Three hours

The figures in the margin indicate full marks for the questions.

1. Answer the following questions : 1×5=5

   (a) Write the applications of queue data structure.

   (b) Give *two* examples of non-linear data structure.

   (c) Write the time complexity of insertion sort.

   (d) What is level order traversal of a tree ?

   (e) What is recursive function ?

2. Answer the following questions :

(a) Write the address translations for row-major ordering and column-major ordering of the dimensional arrays. 5

(b) What is circular queue? Write the advantages of using circular queue using array representation with a suitable example. 1+4=5

(c) What are stack overflow and stack underflow? Give example. 5

(d) Evaluate the given postfix expressions using stack : 5

(i) 7 8 + 3 2 + /

(ii) 2 3 1 * + 9 –

(e) Given expression

$$E = ((2+3)-(5*4))\%((6+7)*(2+5))$$

Construct the corresponding binary tree and also give the post order traversal of the same tree. 5

OR

Write the differences between static and dynamic memory allocation. 5

3. Answer the following questions :

(a) Write the PUSH and POP operation in a stack data structure. 5

(b) Sort the following list using quick sort and show the array after every pass : 5

10, 6, 5, 19, 10, 13, 4

(c) Write the algorithm for merge sort. 5

OR

Write the algorithm for quick sort Algorithm. 5

4. Answer the following questions :

(a) What is doubly linked list ? Write the advantages of doubly linked list over singly linked list. Write the function to insert a node at end in a doubly linked list. 3+3=6

(b) Construct a binary search tree using the following data elements and give the post order traversal of the tree : 2+2=4

75, 43, 9, 85, 76, 14, 25, 90

(c) Write a program to implement insertion sort. 5

OR

Write the function to implement enqueue operation in a circular queue using array representation. 5

5. Write short notes on : ***(any four)*** 4×5=20

(a) Non-primitive data structure

(b) Complexity of algorithms

(c) Heap sort

(d) In-order traversal

(e) Binary search

(f) Memory representation.

DATA STRUCTURE AND ALGORITHMS

Paper : 2·1

(Old Syllabus)

1. Answer the following questions : 1×5=5

   (a) Give *two* examples of stack data structure.

   (b) Give examples of non-linear data structure.

   (c) What is a leaf node ?

   (d) Which data structure is applied for recursive function ?

   (e) Write the time complexity of merge sort.

2. Answer the following questions : ***(any five)*** 3×5=15

   (a) Differentiate between internal sorting and external sorting.

   (b) Explain stack overflow and stack underflow.

(c) What is linked list and what are various types of linked lists? Write the advantages of linked lists over array.

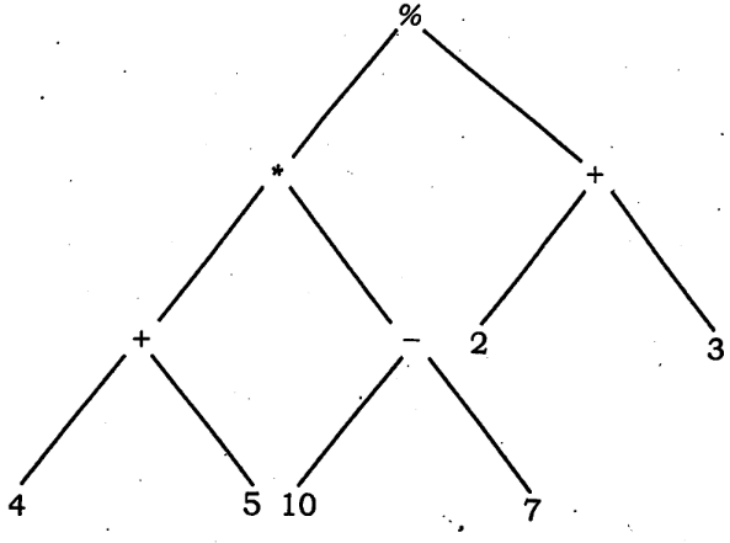
(d) Evaluate the given postfix expressions using stack:

7 8 + 3 2 + /

2 3 1 * + 9 –

(e) Write the algorithm for DEQUEUE operation in a circular queue.

(f) Give the inorder and pre-order traversal of the tree given below:



3. *(a)* Write the algorithm to convert an infix expression into postfix expression. 5

OR

Write the algorithm to implement quick sort.

(b) Sort the following list using selection sort and show the array after every pass : 5

10, 6, 2, 11, 9, 13, 4

(c) What is 2D array ? Write the address translation function for the row-major ordering and column-major ordering in a 2D array. 1+4=5

(d) What is doubly linked list ? Write the functions for the following in a doubly linked list : 1+5=6

(i) insert a node at end

(ii) delete a node at beginning.

(e) What is binary search tree ? Write the function to insert a node in a binary search tree. 1+4=5

(f) Write the function to insert a node at end in a singly linked list. 4

OR

Write the function to enqueue an element in a queue using linked list representation. 4

(g) Write the algorithm for BFS. 5

4. Write short notes on : ***(any five)*** 5×5=25

(a) Heap sort

(b) Circular linked list

(c) Memory representation

(d) Hashing technique

(e) Complexity of algorithms

(f) Tree traversal.