

Roll No.

M.Sc.IT-10 (Master of Science in Information Technology)

Second Semester, Examination-2014

MIT-2001

Data Structure & Algorithm

Time Allowed : Three Hours

Maximum Marks : 60

Note : This paper is of sixty (60) marks divided into three (03) sections. Learners are required to attempt the questions contained in these sections according to the detailed instructions given therein.

Section - A

(Long answer type Questions)

Note : Section 'A' contains four (04) long-answer-type questions of fifteen (15) marks each. Learners are required to answer any two (02) questions only. (2×15=30)

1. (a) What is a data structure and what are the differences between data type, abstract data type and data structure?
(b) Define algorithm and design an algorithm to find out the total number of even and odd numbers in a list of 100 numbers.
2. (a) Explain the following with the help of example:
(i) Pointer to structure (ii) Pointer in structure
(b) What is structure? How is different from an array? Explain different types of structure declaration with examples and its initializations.
3. (a) Write the functions for following operations on linked list:
(i) Insert (ii) Delete
(iii) Modify

- (b) Write an expression tree for the following postfix expression
 $ab+cd-*ef+/.$
4. (a) Differentiate the following:
- (i) Static and dynamic data structure
 - (ii) Linear and Non linear data structure.
- (b) Describe stack and write algorithms for Push and POP operations.

Section - B

(Short answer type Questions)

Note : Section 'B' contains eight (08) short-answer-type questions of five (05) marks each. Learners are required to answer any four (04) questions only. (4×5=20)

1. Convert the following to infix
 - (i) $AB-DE*+$
 - (ii) $-*AB*ED$
2. Explain the following terms:
 - (a) Graph
 - (b) Directed Graph
 - (c) Simple Path
 - (d) Sub Graph
 - (e) Undirected Graph
3. Explain the following:
 - i) Circular Linked List
 - ii) Doubly Linked List.Using suitable example.
4. What are the advantages doubly linked list over singly linked list.
5. What are the applications of tree, write some examples.
6. Explain the following terms:
 - i. child
 - ii. Siblings

- iii. Ancestor
 - iv. Descendents
 - v. Root Node
7. What are Header nodes? How do header nodes help to simplify the operation of Circular Doubly Linked List?
8. Define the algorithms for infix to polish notations.

Section - C

(Objective type Questions)

Note : Section 'C' contains ten (10) objective-type questions of one (01) mark each. All the questions of this section are compulsory. (10×1=10)

1. The data structure required to evaluate a postfix expression is
- (A) queue
 - (B) stack
 - (C) array
 - (D) linked-list
2. What data structure would you mostly likely see in a nonrecursive implementation of a recursive algorithm?
- (A) Stack
 - (B) Linked list
 - (C) Queue
 - (D) Trees
3. Which of the following sorting methods would be most suitable for sorting a list which is almost sorted.
- (A) Bubble Sort
 - (B) Insertion Sort
 - (C) Selection Sort
 - (D) Quick Sort
4. The process of accessing data stored in a serial access memory is similar to manipulating data on a
- (A) heap
 - (B) queue
 - (C) stack
 - (D) binary tree

5. A linear collection of data elements where the linear node is given by means of pointer is called
(A) linked list (B) node list
(C) primitive list (D) None of these
6. Representation of data structure in memory is known as:
(A) recursive (B) abstract data type
(C) storage structure (D) file structure
7. Memory allocation at the compile time is known as
(A) Static memory allocation
(B) Dynamic memory allocation
(C) Paging
(D) None of the above
8. Get the value of most recently inserted node and delete the node refers to operation
(A) POP (B) PUSH
(C) EMPTY (D) None of the above
9. "FRONT=REAR" pointer refers to empty
(A) Stack (B) Queue
(C) Array (D) None of the above
10. Data Structure —
(A) May be helpful to develop efficient algorithms in different phases of data processing
(B) need not give relationship between data items.
(C) is programming language dependent
(D) None of the above