



DBB-003-007202

Seat No. _____

M. C. A. (Sem. II) (CBCS) Examination

May / June – 2015

CCA - 2002: Data structure and Algorithm

Faculty Code : 003

Subject Code : 007202

Time : $2\frac{1}{2}$ Hours]

[Total Marks : 70

1 Answer the following multiple choice questions : 15

- (1) _____ Is a pile in which items are added at one end and removed from the other.
 - (a) Stack
 - (b) Queue
 - (c) List
 - (d) None of the above.
- (2) Sparse matrices have _____.
 - (a) Many zero entries
 - (b) Many non-zero entries
 - (c) Higher dimension
 - (d) None of the above.
- (3) The postfix expression for $* + a b - c d$ is?
 - (a) $ab + cd - *$
 - (b) $ab cd + - *$
 - (c) $ab + cd * -$
 - (d) $ab + - cd *$
- (4) In a priority queue insertion and deletion takes place at
 - (a) front, rear end
 - (b) only at rear end
 - (c) only at front end
 - (d) any position
- (5) Sort which compares adjacent elements in a list and switches where necessary is _____.
 - (a) Insertion sort
 - (b) Heap sort
 - (c) Bubble sort
 - (d) Quick sort

- (6) Given a binary tree, which traversal type will print values in the nodes in sorted order
- (a) Preorder (b) Postorder
- (c) Inorder (d) None of the above.
- (7) Each node in a linked list must contain at least ____.
- (a) Three field (b) Two field
- (c) One field (d) Four field
- (8) To implement Sparse matrix dynamically, the following data structure is used
- (a) Tree (b) Graph
- (c) Priority Queue (d) Linked list
- (9) Which data structure is useful to convert infix notation to postfix notation
- (a) Linear list (b) Stack
- (c) Tree (d) Queue
- (10) Applications of Linked List are
- (a) Simulation , event driven systems
- (b) Postfix and prefix manipulations
- (c) Dictionary systems, polynomial manipulations
- (d) Fixed block storage allocation, garbage collection
- (11) Which of the following name does not relate to stacks
- (a) FIFO list (b) LIFO list
- (c) Piles (d) Push-down-list

- (12) Binary search algorithm can be applied to
- (a) Sorted linked list
 - (b) Sorted binary trees
 - (c) Sorted linear array
 - (d) Pointer array
- (13) Which of the following sorting algorithm is of divide-and-conquer type?
- (a) Bubble sort
 - (b) Insertion sort
 - (c) Quick sort
 - (d) All of the above
- (14) Hashing collision resolution techniques are
- (a) Huffman coding, linear hashing
 - (b) Bucket addressing, Huffman coding
 - (c) Chaining, Huffman coding
 - (d) Chaining, Bucket addressing
- (15) Recursive problems are implemented using _____.
- (a) Queue
 - (b) Stack
 - (c) Linked list
 - (d) String

2 Attempt **any five** of the following :

15

- (1) What is primitive data structure?
- (2) State the principle of stack and give its two applications.
- (3) What is array of structure? Explain with brief example.
- (4) What are sparse matrices?
- (5) Define: linked allocation. List applications of linked linear list.
- (6) What is priority queue? How it is different from queue.

- 3** Attempt **any three** of the following : **15**
- (1) What is hashing? List various hashing functions and explain any one in detail.
 - (2) Explain reversing and copy string to other string without using built-in function.
 - (3) What is singly linked linear list? Write algorithm / program to delete node from singly linked linear list from particular location.
 - (4) Explain in detail application of trees.
- 4** Attempt **any two** of the following : **15**
- (1) What is tree? Explain preorder, inorder and postorder traversal of a binary tree taking suitable example.
 - (2) What is queue? How it is different from stack? Explain various operations on queue.
 - (3) What is binary search? State advantage of binary search against sequential search. Write algorithm / program to demonstrate binary search.
- 5** Attempt **any one** of the following : **10**
- (1) List out various sorting techniques. Explain Radix sort and Bubble sort in detail.
 - (2) Define: doubly linked linear list. Explain with example various operations on doubly linked linear list such as creating list, insert node in a list (k^{th} position), delete node (from k^{th} position) from a list.