

Roll No.

97670

**BCA 3rd Semester (New)
Examination – November, 2018**

DATA STRUCTURE-I

Paper : BCA-202

Time : Three Hours]

[Maximum Marks : 80

Before answering the questions, candidates should ensure that they have been supplied the correct and complete question paper. No complaint in this regard, will be entertained after examination.

Note : Attempt *five* questions in all. Question No. 1 is *compulsory* and attempt *four* more questions by selecting *one* question from each Unit. All questions carry equal marks.

1. (a) What is string ?
- (b) Describe the Big-O notation.
- (c) What is doubly linked list ?
- (d) Write the advantages of circular list.
- (e) What is recursion ?
- (f) What is priority queue ?
- (g) Write the properties of binary tree.
- (h) What is a graph ?

UNIT - I

2. (a) What is the need of data structure ? Discuss various types of data structure.
(b) What do you mean by efficiency of an algorithm ? Explain the concept of best case, average case and worst case time complexity.
3. What do you mean by pattern matching ? Explain various patterns matching algorithm by using example.

UNIT - II

4. What is an array ? Discuss the various operations on linear array and write an algorithm for inserting and deleting an element into a linear array.
5. What is the difference between array and linked list ? How can you represent a linked list in memory ? Explain the insertion and deletion operations of linked list by giving suitable example.

UNIT - III

6. (a) What are the basic operations performed on stack ? Write down the steps to perform these operations.
(b) What is postfix notation ? Explain the method of evaluating postfix expression by giving suitable example.
7. What are queues ? How are queues implemented in memory ? What are the various queue operations ? Write algorithm for each.

UNIT - IV

8. What is binary tree and strictly binary tree ? Explain the various methods of representation a binary tree in memory.
9. What is meant by traversal of a graph ? Discuss the breadth first and depth first traversal techniques with the help of example.