K25P 2251

Reg. No. : .....

Name : .....

Second Semester M.C.A. Degree (C.B.S.S. – Reg./Supple./Imp.) Examination, May 2025
(2021 Admission Onwards)
MCA2 C01 : ALGORITHMS AND DATA STRUCTURES

Time : 3 Hours

Max. Marks : 60

SECTION – A

Answer **all** questions. **Each** Question carries **two** marks.

1. What is an algorithm ? Explain its characteristics.
2. What do you mean by backtracking ?
3. Explain time complexity and space complexity.
4. Differentiate between NP Hard and NP complete problems.
5. Define arrays.
6. List out the applications of stack.
7. What is a tree ? Represent a binary tree.
8. Explain BST.
9. What is a graph ? List out the applications of graph.
10. Explain directed graph and acyclic graph.

(10×2=20)

P.T.O.

SECTION – B

Answer **all** questions. **Each** question carries **eight** marks.

11. a) Explain the steps involved in the development of an algorithm with a neat diagram.

OR

b) Explain any three asymptotic notations used to express the complexity of algorithm with the help of suitable examples.

12. a) What do you mean by priority queue ? Write an algorithm to insert an element into a priority queue.

OR

b) Write an algorithm to convert an infix expression into its equivalent postfix expression. Convert the expression ((A /(B – D + E)) *(F – G) *H) to postfix form. Show each step in the conversion including the stack contents.

13. a) Explain PUSH and POP of operations of a stack with an example.

OR

b) What do you mean by a circular linked list ? Write an algorithm to perform insert and delete operations on a circular linked list.

14. a) Write and discuss algorithm to insert an element to binary search tree. Show the structure of the binary search tree after adding each of the following values in that order : 2, 5, 1, 7, 10, 9, 11, 6.

OR

b) Explain all types of binary tree. Describe preorder, inorder, postorder tree traversal of tree with the help of an example.

15. a) Illustrate depth first search algorithm.

OR

b) Explain with an example binary search on a linear array. **(5×8=40)**