# BRAINWARE UNIVERSITY



**Term End Examination 2024-2025**
**Programme – B.Tech.(CSE)-2023**
**Course Name – Data Structure & Algorithms**
**Course Code - ESCG302**
**( Semester III )**

**Full Marks : 60** **Time : 2:30 Hours**

[The figure in the margin indicates full marks. Candidates are required to give their answers in their own words as far as practicable.]

## Group-A

(Multiple Choice Type Question) 1 x 15=15

1. *Choose the correct alternative from the following :*

(i) Finding the location of a given item in a collection of items is defined as
a) Finding b) Searching
c) Mining d) Find

(ii) Select the worst case time complexity of linear search algorithm?
a) O(N) b) O(logn)
c) O(nlogn) d) O(1)

(iii) Select the best case complexity of bubble sort improve version
a) O b) O(n logn)
c) O(n2) d) O(n3)

(iv) Choose the correct one when an insertion operation can take inside the circular Queue
a) Rear =Rear -1 b) Rear =Rear +1
c) Font =Font+1 d) Font =Font-1

(v) State the formula to find a parent node of a tree be located in an array
a) (i+1)/2 b) (i-1)/2
c) i/2 d) 2i/2

(vi) Compute the array int x [10][20] stores how many elements?
a) 230 elements b) 200 elements
c) 8000 elements d) any number of elements

(vii) Select the correct options. At any time the position of the element to be inserted will be calculated by the relation
a) Rear = (Rear + 1) % SIZE b) Rear = (Font + 1) % SIZE
c) Font = (Rear + 1) % SIZE d) Font = (Font + 1) % SIZE

(viii) Compute is the time complexity of a program to reverse a linked list?
a) O(n) b) O(1)
c) O(n^2) d) O(logn)

(ix) solve the space complexity for deleting a linked list?

a) o(1)
b) o(n)
c) o(n^2)
d) none of these

(x) Choose the best arrangement of the array elements after second pass using insertion sort for [34, 8, 64, 51, 32, 21]

a) 8, 21, 32, 34, 51, 64
b) 8, 32, 34, 51, 64, 21
c) 8, 34, 51, 64, 32, 21
d) 8, 34, 64, 51, 32, 21

(xi) Select the best option for external sorting algorithm

a) Algorithm that uses tape or disk during the sort
b) Algorithm that uses main memory during the sort
c) Algorithm that involves swapping
d) Algorithm that are considered 'in place'

(xii) Identify the correct option for depth First Search

a) DFS can be implemented using an explicit stack or recursion (which uses the call stack).
b) is equivalent to Pre-order traversal of binary tree
c) is equivalent to In-order traversal of binary tree
d) is equivalent to Level-order traversal of binary tree

(xiii) Choose the correct data structure for fast look-up and deletion operations.

a) Array
b) Hash Table
c) Queue
d) Tree

(xiv) State the meaning of 'insertion operation' in data structures.

a) Adding an element in the middle
b) Removing an element from the front
c) Adding an element in a specific location
d) Replacing an element with a new one

(xv) List the types of expressions that can be evaluated using stacks.

a) Infix
b) Prefix
c) Postfix
d) In-order

## Group-B

(Short Answer Type Questions) 3 x 5=15

2. Describe the different types of arrays (one-dimensional, two-dimensional) (3)
3. Define Time Complexity. (3)
4. Define the terms: Degree of a vertex and Path in a graph (3)
5. Compare BFS and DFS algorithms. (3)
6. Explain the concept of a push algorithm in a stack. (3)

**OR**

Explain the concept of a pop algorithm in a stack. (3)

## Group-C

(Long Answer Type Questions) 5 x 6=30

7. Compare the performance of binary search trees (BSTs) and AVL trees in terms of insertion, deletion, and searching operations. (5)
8. Define the algorithm for BFS using an example. (5)
9. Evaluate the importance of asymptotic notations in algorithm analysis and discuss how they guide the selection of efficient algorithms. (5)
10. Explain the reason behind Quick Sort considered better in practice than other comparison-based sorting algorithms. (5)
11. Compare and contrast Simple Queue and Circular Queue. (5)

12. Classify the various sorting algorithms (Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort, Radix Sort) according to their stability, time complexity, and space complexity. (5)

**OR**

Explain different type of probing in hashing with example. (5)



****************************************