BRAINWARE UNIVERSITY

Term End Examination 2023-2024
Programme – Dip.EE-2021/Dip.CSE-2022
Course Name – Data Structure and Algorithm/Data Structures
Course Code - DEE304/DCSE-PC301
(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) Identify the data structure that follows the last-in, first-out (LIFO) principle.

a) Queue b) Stack
c) Linked list d) Tree

(ii) Name the process of finding an element in a data structure.

a) Insertion b) Deletion
c) Searching d) Sorting

(iii) Select the term that describes a data structure\'s ability to store multiple elements of the same type.

a) Polymorphism b) Encapsulation
c) Abstraction d) Homogeneity

(iv) Name the data structure that represents a linear collection of elements with constant-time insertions and deletions at both ends.

a) Queue b) Stack
c) Linked list d) Array

(v) Select the sorting algorithm that is considered an in-place, stable sorting algorithm that builds the final sorted array one item at a time.

a) Merge Sort b) Quick Sort
c) Bubble Sort d) Selection Sort

(vi) Trace the sorting algorithm that selects a \"pivot\" element and partitions the array into two subarrays: elements less than the pivot and elements greater than the pivot.

a) Merge Sort b) Heap Sort
c) Bubble Sort d) Quick Sort

(vii) Cite the sorting algorithm that has a worst-case time complexity of O(n^2) but is efficient for small datasets or mostly sorted data.

a) Merge Sort b) Quick Sort
c) Selection Sort d) Bubble Sort

(viii) Identify the sorting algorithm that is based on the idea of maintaining a \"heap\" data structure to sort the elements.

a) Quick Sort b) Bubble Sort
c) Merge Sort d) Heap Sort

(ix) Name the data structure that allows elements to be accessed directly using an index.

a) Stack b) Queue
c) Linked list d) Array

(x) Identify the sorting algorithm that can be efficient for small datasets or nearly sorted data but may perform poorly on large, unsorted datasets.

a) Heap Sort b) Quick Sort
c) Merge Sort d) Bubble Sort

(xi) Identify the linear data structure that follows the Last-In, First-Out (LIFO) principle.

a) Queue b) Linked List
c) Stack d) Array

(xii) Name the fundamental operation that adds an element to the top of a stack.

a) Push b) Pop
c) Peek d) Enqueue

(xiii) Select the abstract data type (ADT) that accurately represents a stack\'s behavior.

a) List b) Set
c) Stack d) Dictionary

(xiv) Choose the operation that removes and returns the top element of a stack.

a) Push b) Pop
c) Peek d) Enqueue

(xv) Select the operation that retrieves the number of elements in a stack.

a) Push b) Pop
c) Size d) Enqueue

Group-B

(Short Answer Type Questions) 3 x 5=15

2. Describe the primary purpose of data structures in computer science. (3)
3. Select one advantage of using a linked list over an array for dynamic memory allocation. (3)
4. Discuss the key requirement for binary search to work effectively. (3)
5. Explain how binary search narrows down the search space with each comparison. (3)
6. Classify binary trees as a type of hierarchical data structure. (3)

OR

Explain the key difference between a binary search tree and an AVL tree in terms of balancing. (3)

Group-C

(Long Answer Type Questions) 5 x 6=30

7. Define a linked list and compare it to an array in terms of memory allocation and dynamic sizing. Explain the advantages of using a singly linked list over an array. (5)
8. Describe the structure of a singly linked list and its nodes. Explain how insertions and deletions are performed in a singly linked list. (5)
9. Describe the process of linear search and illustrate it with an example. Discuss the scenarios where linear search is an appropriate choice. (5)
10. Analyze the key principles behind the bubble sort algorithm and discuss its performance on large datasets. How does it differ from quicksort? (5)
11. Classify sorting algorithms into comparison-based and non-comparison-based categories. Explain the fundamental differences between these two categories. (5)
12. Consider a scenario where external sorting is necessary due to limited memory resources. Decide on the most appropriate nonlinear data structure for sorting large datasets efficiently in an external memory environment. (5)

OR

Compare and contrast the performance of a binary search tree and an AVL tree in terms of insertion, deletion, and searching operations. Consider scenarios where one may be preferred over the other. (5)

****