

Bachelor of Computer Applications (BCA)
Semester 4 - Data Structures and Algorithms

Time: 3 hours

Maximum Marks: 100

Section A: Short Answer Questions (5 x 4 = 20 marks)

Answer all questions. Each question carries 4 marks.

1. Define the term 'Abstract Data Type' and give an example.
2. What is the difference between a stack and a queue? Provide a real-world analogy for each.
3. Explain the concept of recursion. Write a recursive function to calculate the factorial of a number.
4. What is a binary search tree? How does it differ from a binary tree?
5. Define Big O notation. Why is it important in algorithm analysis?

Section B: Programming Questions (3 x 10 = 30 marks)

Answer all questions. Each question carries 10 marks.

6. Implement a stack data structure using an array in any programming language of your choice. Include push, pop, and display functions.
7. Write a program to reverse a linked list. Explain the algorithm you used.
8. Implement the bubble sort algorithm to sort an array of integers in ascending order. Analyze its time complexity.

Section C: Long Answer Questions (2 x 15 = 30 marks)

Answer any two questions. Each question carries 15 marks.

9. Explain the working of a hash table. Discuss different collision resolution techniques with examples.
10. Describe the Depth-First Search (DFS) and Breadth-First Search (BFS) algorithms for graph traversal. Compare and contrast their applications.
11. Explain the concept of dynamic programming. Solve the "0/1 Knapsack Problem" using dynamic programming approach.

Section D: Case Study (1 x 20 = 20 marks)

Answer the following case study.

12. You are tasked with developing a system for a library to manage book borrowing and returns. The system should be able to:

- a) Add new books to the library inventory
- b) Allow members to borrow books
- c) Process book returns
- d) Display the list of available books
- e) Show the list of books borrowed by a specific member

Design an efficient data structure to store book and member information. Explain your choice of data structure and provide a high-level algorithm for each of the above operations. Discuss any potential challenges in implementing this system and how you would address them.

Instructions:

- Read all questions carefully before attempting.
- Provide clear and concise answers with proper explanations where required.
- You may use pseudocode or any standard programming language for coding questions.
- Neat diagrams must be drawn wherever necessary.