Assignment: Data Structure and Algorithm
Total Marks: 100 | Level: BCS 3rd Year, 6th Semester

Table of Contents

Section A: Theoretical Questions (30 Marks)

Answer three questions. (3 × 10 = 30 marks)

1. Define Time and Space Complexity.
    - Explain best, worst, and average-case time complexity with examples of common algorithms (e.g., linear search, binary search, bubble sort).
2. Describe how a stack is used in expression evaluation.
    - Compare Linear Queue and Circular Queue.
    - Provide scenarios where circular queue is preferred over linear queue.
3. Differentiate between singly, doubly, and circular linked lists.
    - Discuss the advantages of using linked lists over arrays.
    - Provide basic insertion and deletion operations.
4. Define recursion.
    - Compare recursion with iteration using the example of factorial.
    - Discuss the pros and cons of using recursion.

Section B: Numerical / Problem-Solving (30 Marks)

Answer three questions. (3 × 10 = 30 marks)

1. Convert the following infix expression into postfix and prefix notation:
   (A + B) * (C - D) / E
    - Evaluate the postfix expression for A=2, B=3, C=5, D=1, E=2.
2. Given the array:
   A = [8, 5, 3, 9, 1]

- Sort the array using Bubble Sort and show all steps.
- Calculate the total number of comparisons and swaps.

3. Given a sorted array:
   A = [2, 4, 6, 8, 10, 12, 14, 16, 18]
   - Search for element 10 using Binary Search.
   - Show the steps and calculate its time complexity.
4. Given the following undirected weighted graph:
5. A --2-- B
6. A --3-- C
7. B --1-- C
8. B --4-- D
9. C --5-- D
   - Use Kruskal's Algorithm to find the Minimum Spanning Tree.
   - Show all steps and the final MST.

Section C: Programming / Implementation (40 Marks)

Answer any two questions. (2 × 20 = 40 marks)

1. Write a program (in C) to implement a stack using an array.
   - Include functions for push, pop, peek, and display.
   - Demonstrate stack operations with sample inputs.
2. Write a program to implement a singly linked list with the following operations:
   - Insert at beginning
   - Insert at end
   - Delete a node from a given position
   - Display the list
   - Provide sample input and output
3. Implement the Tower of Hanoi problem for n disks using recursion.
   - Provide source code
   - Show the output for n = 3

4. Implement a hash table using linear probing for collision resolution.
    - Accept keys from the user and store them in a fixed-size hash table
    - Show the hash function used
    - Provide sample input and the resulting table

Texas College of Management & IT

Bachelor of Computer Science (Network Technology & Cyber Security)

Project Title: _____

Submitted By:

Name: _____

LCID: _____

Semester: 3rd Year, 6th Semester

Submitted To:

Instructor Name: _____

Course: _____

Date of Submission: _____